The Linux Printing Usage HOWTO

by Mark Komarinski <markk@auratek.com> v1.11, 2 February 1996

1 Introduction

This document describes how to use the line printer spooling system provided with the Linux operating system. This HOWTO is the supplementary document to the Linux Printing Setup HOWTO, which discusses the installation and setup of the Linux printing system. The material presented in this HOWTO should be equally relevent for all flavors of the BSD operating system in addition to the Linux operating system.

1.1 Linux Printing HOWTO History

This version of the Linux Printing HOWTO is a complete rewrite of the one originally written by Grant Taylor <grant@god.tufts.edu> and Brian McCauley <B.A.McCauley@bham.ac.uk>. I have tried to keep with the coverage of material presented by Grant and Brian's HOWTO, but I have drastically modified the style of presentation and the depth of material covered. I feel that this makes the HOWTO more complete and easier to read. I can only hope that you agree.

1.2 Version History

v1.11

- new maintainter!
- Added lpc info
- Added some info for troubleshooting
- A start on printing graphics files!

v1.1

- revised some of the wording
- developed section on PostScript printing
- attempted to clarify some of the examples 8-)
- fleshed the discussion of the basic Linux printing utilities

v1.0

- initial public release of the Printing Usage HOWTO

1.3 Copyrights and Trademarks

1.4 Downloading the Linux Printing HOWTOs

I recommend that if you want to print a copy of this HOWTO that you download the PostScript version. It is formatted in a fashion that is aesthetically appealing and easier to read. You can get the PostScript version from one of the many Linux distribution sites (such as SunSITE ??).

1.5 Feedback

Questions, comments, or corrections for this HOWTO may be directed to <markk@auratek.com>.

1.6 Acknowledgments

Thanks go out to all of the people who took the time to read the alpha version of this HOWTO and respond with many helpful comments and suggestions—some of you may see your comments reflected in the version.

I would also like to thank Grant and Brian for the loads of information that they have given me during the transition of authorship.

2 Printing Under Linux

This section discusses how to print files, examine the print queue, remove jobs from the print queue, format files before printing them, and configure your printing environment.

2.1 History of Linux Printing

The Linux printing system—the *lp* system—is a port of the source code written by the Regents of the University of California for the Berkeley Software Distribution version of the UNIX operating system.

2.2 Printing a File Using *lpr*

By far, the most simplistic way to print in the Linux operating system is to send the file to be printed directly to the printing device. One way to do this is to use the *cat* command. As the `root` user, one could do something like

```
# cat thesis.txt > /dev/lp
```

In this case, `/dev/lp` is a symbolic link to the actual printing device—be it a dot-matrix, laser printer, typesetter, or plotter. (See *ln*(1) for more information on symbolic links.)

For the purpose of security, only the `root` user and users in the same group as the print daemon are able to write directly to the printer. This is why commands such as *lpr*, *lprm*, and *lpq* have to be used to access the printer.

Because of this, users have to use *lpr* to print a file. The *lpr* command takes care of all the initial work needed to print the file, and then it hands control over to another program, *lpd*, the line printing daemon. The line printing daemon then tells the printer how to print the file.

When *lpr* is executed, it first copies the specified file to a certain directory (the spool directory) where the file remains until *lpd* prints it. Once *lpd* is told that there is a file to print, it will spawn a copy of itself (what we programmers call forking). This copy will print our file while the original copy waits for more requests. This allows for multiple jobs to be queued at once.

The syntax of *lpr*(1) is a very familiar one,

```
$ lpr [ options ] [ filename ... ]
```

If `filename` is not specified, *lpr* expects input to come from standard input (usually the keyboard, or another program's output). This enables the user to redirect a command's output to the print spooler. As such,

```
$ cat thesis.txt | lpr
```

or,

```
$ pr -l60 thesis.txt | lpr
```

The *lpr* command accepts several command-line arguments that allow a user to control how it works. Some of the most widely used arguments are: -**P**printer specifies the printer to use, -**h** suppresses printing of the burst page, -**s** creates a symbolic link instead of copying the file to the spool directory (useful for large files), and -**#**num specifies the number of copies to print. An example interaction with *lpr* might be something like

```
$ lpr -#2 -sP dj thesis.txt
```

This command will create a symbolic link to the file `thesis.txt` in the spool directory for the printer named *dj*, where it would be processed by *lpd*. It would then print a second copy of `thesis.txt`.

For a listing of all the options that *lpr* will recognize, see *lpr*(1).

2.3 Viewing the Print Queue with *lpq*

To view the contents of the print queue, use the *lpq* command. Issued without arguments, it returns the contents of the default printer's queue.

The returned output of *lpq* can be useful for many purposes.

```
$ lpq
lp is ready and printing
Rank   Owner      Job  Files                             Total Size
active mwf        31   thesis.txt                        682048 bytes
```

2.4 Canceling a Print Job Using *lprm*

Another useful feature of any printing system is the ability to cancel a job that has been previously queued. To do this, use *lprm*.

```
$ lprm -
```

The above command cancels all of the print jobs that are owned by the user who issued the command. A single print job can be canceled by first getting the job number as reported by *lpq* and then giving that number to *lprm*. For example,

```
$ lprm 31
```

would cancel job 31 (`thesis.txt`) on the default printer.

2.5 Controlling the lpd program with *lpc*

The *lpc*(8) program is used to control the printers that lpd serves. you can enable or disable a printer or its queues, rearrange entries within a queue, and get a status report on the printers and their queues. Lpc is mostly used in a setup where there are multiple printers hanging off one machine.

```
$ lpc
```

The above will start the lpc program. By default, this enters you into an interactive mode, and you can begin issuing commands. The other option is to issue an lpc command on the command line.

```
$ lpc status all
```

A list of the available commands are in the *lpd* man page, but here are a few of the major commands you'll want to know about. Any commands marked with *option* can either be a printer name (lp, print, etc) or the keyword `all`, which means all printers.

- disable *option* - prevents any new printer job from being entered
- down *option* - disables all printing on the printer
- enable *option* - allow new jobs to enter the print queue
- quit (or exit) - leave lpc
- restart *option* - restarts lpd for that printer
- status *option* - print status of printer
- up *option* - enable everything and start a new lpd

3 Miscellaneous Items

This section discusses some of the miscellaneous things that you may want to know about printing under Linux.

3.1 Formatting Before Printing

Since most ASCII files are not formatted for printing, it is useful to format them in some way before they are actually printed. This may include putting a title and page number on each page, setting the margins, double spacing, indenting, or printing a file in multiple columns. A common way to do this is to use a print preprocessor such as *pr*.

```
$ pr +4 -d -h"Ph.D. Thesis, 2nd Draft" -l60 thesis.txt | lpr
```

In the above example, *pr* would take the file `thesis.txt` and skip the first three pages (+4), set the page length to sixty lines (-l60), double space the output (-d), and add the phrase "Ph.D. Thesis, 2nd Draft" to the top of each page (-h). *Lpr* would then queue *pr*'s output. See its on-line manual page for more information on using *pr*.

3.2 The PRINTER Environment Variables

All of the commands in the Linux printing system accept the -**P** option. This option allows the user to specify which printer to use for output. If a user doesn't specify which printer to use, then the default printer will be assumed as the output device.

Instead of having to specify a printer to use every time that you print, you can set the PRINTER environment variable to the name of the printer that you want to use. This is accomplished in different ways for each shell. For *bash* you can do this with

```
$ PRINTER="printer_name"; export PRINTER
```

and *csh*, you can do it with

```
% setenv PRINTER "printer_name"
```

These commands can be placed in your login scripts (.profile for *bash*, or .cshrc for *csh*), or issued on the command-line. (See *bash*(1) and *csh*(1) for more information on environment variables.)

3.3 Printing graphics files

Printing graphics files through a printer usually depends on the kind of graphics you're converting, and the kind of printer you want to send to. Dot matrix is usually out of the question due to differences in the way dot-matrix handles graphics. Your best bet in this situation is to see if your printer is compatable with an Epson or an IBM ProPrinter, then convert the graphics file to PostScript, then use Ghostscript (see next section) to print the graphics.

If you have a laser printer, things are a bit easier since many are compatable with PCL. This now gives you a few options. Some programs may output directly in PCL. If not, programs like *NetPBM* can convert into PCL. Last option is to use ghostscript (see next section).

Your absolutely best option is to install packages like NetPBM and Ghostscript then installing a magic filter to process the graphics files automagically.

3.4 Printing PostScript files

Printing PostScript files on a printer that has a PostScript interpreter is simple; just use *lpr*, and the printer will take care of all of the details for you. For those of us that don't have printers with PostScript capabilities, we have to resort to other means. Luckily, there are programs available that can make sense of PostScript, and translate it into a language that most printers will understand. Probably the most well known of these programs is Ghostscript.

Ghostscript's responsibility is to convert all of the descriptions in a PostScript file to commands that the printer will understand. To print a PostScript file using Ghostscript, you might do something like

```
$ gs -dSAFER -dNOPAUSE -sDEVICE=deskjet -sOutputFile=|lpr thesis.ps
```

Notice in the above example that we are actually piping the output of Ghostscript to the *lpr* command by using the **-sOutputFile** option.

Ghostview is an interface to Ghostscript for the X Window System. It allows you to preview a PostScript file before you print it. Ghostview and Ghostscript can both be swiped from ??.

3.5 Printing TeX files

One of the easiest ways to print TeX files is to convert them to PostScript and then print them using Ghostscript. To do this, you first need to convert them from TeX to a format known as DVI (which stands for device-independent). You can do this with the *tex*(1) command. Then you need to convert the DVI file to a PostScript file using *dvips*. All of this would look like the following when typed in.

```
$ tex thesis.tex
$ dvips thesis.dvi
```

Now you are ready to print the resulting PostScript file as described above.

3.6 Printing `troff` formatted files

```
$ groff -Tascii thesis.tr | lpr
```

or, if you prefer,

```
$ groff thesis.tr > thesis.ps
```

and then print the PostScript file as described above.

4 Answers to Frequently Asked Questions

Q1. How do I prevent the staircase effect?

A1. The staircase effect is caused by the way some printers expect lines to be terminated. Some printers want lines that end with a carriage-return/line-feed sequence (DOS-style) instead of the line-feed sequence used for UNIX-type systems. The easiest way to fix this is to see if your printer can switch between the two styles somehow—either by flipping a DIP switch, or by sending an escape sequence at the start of each print job. To do the latter, you need to create a filter (see Q2 and $Foster95b$).

A quick fix is to use a filter on the command-line. An example of this might be

```
$ cat thesis.txt | todos | lpr
```

Q2. What is a filter?

A2. A filter is a program that reads from standard input (*stdin*), performs some action on this input, and writes to standard output (*stdout*). Filters are used for a lot of things, including text processing.

Q3. What is a *magic* filter?

A3. A magic filter is a filter that performs an action based on a file's type. For example, if the file is a plain, text file, it would simply print the file using the normal methods. If the file is a PostScript file, or any other format, it would print it using another method (ghostscript). Two examples of this is magicfilter and APSfilter. One caveat of these filters is that the appropriate programs have to be installed before you install the filter.

5 Troubleshooting

This section covers some common things that can go wrong with your printing system.

If your printer doesn't work:

- Do other print jobs work? (application)
- Is lpd running? (check it using lpc) (print controller)
- Can root send something directly to the printer? (print services)
- Can you print from DOS? (cable/printer problem?)

Answering these questions can help find a solution.

Send other suggestions for this section to <markk@auratek.com>.

6 References

This is a section of references on the Linux printing system. I have tried to keep the references section of this HOWTO as focused as possible. If you feel that I have forgotten a significant reference work, please do not hesitate to contact me.

Before you post your question to a USENET group, consider the following:

- Is the printer accepting jobs? (Use *lpc*(8) to verify.)
- Is the answer to your question covered in this HOWTO, or the Printing Setup HOWTO?

If any of the above are true, you may want to think twice before you post your question. And, when you do finally post to a newsgroup, try to include pertinent information. Try not to just say something like, "I'm having trouble with lpr, please help." These types of posts will most definitely be ignored by many. Also try to include the kernel version that you're running, how the error occured, and, if any, the specific error message that the system returned.

Foster, Matt. *Linux Printing Setup HOWTO*

the supplement to this HOWTO; covers setting up, and configuring the print software

Welsh, Matt. *Linux Installation and Getting Started*

an excellent introductory text for the beginning Linux user

On-Line Manual Pages

- *cat*(1) concatenate and print files
- *dvips*(1) convert a TeX DVI file to PostScript
- *ghostview*(1) view PostScript documents using Ghostscript
- *groff*(1) front-end for the groff document formatting system
- *gs*(1) Ghostscript interpreter/viewer
- *lpc*(8) line printer control program
- *lpd*(8) line printer spooler daemon
- *lpq*(1) spool queue examination program
- *lpr*(1) off-line printer
- *lprm*(1) remove jobs from the line printer spooling queue
- *pr*(1) convert text files for printing
- *tex*(1) text formatting and typesetting

USENET newsgroups

- `comp.os.linux.*` a plethora of information on Linux
- `comp.unix.*` discussions relating to the UNIX operating system