

lpd Printing With FreeBSD

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
	2011-05-30		WB

Contents

1	Introduction	1
2	Printing Directly	1
3	Creating A Spool Directory	1
4	Enabling <i>lpd</i>	1
5	Parallel Or USB Port	2
6	Adding A Filter	3
7	Smart Filters	3
8	Network Printers	4
9	Different Queues For Different Things	5
10	Conclusion	5

© 2011 Warren Block

Last updated 2011-05-30

Available in [HTML](#) or [PDF](#). Links to all my articles [here](#). Created with [AsciiDoc](#).

Using BSD *lpd* for local and network printers on FreeBSD.

1 Introduction

lpd(8) is the standard FreeBSD print spooler. It accepts print jobs from local or remote users and spools them to printers defined in */etc/printcap*.

2 Printing Directly

A spooler is not required. Sometimes it's simpler to just print data directly to a device:

```
% cat myfile.txt > /dev/lpt0
```

<i>/dev/lpt0</i>	the standard parallel port
<i>/dev/ulpt0</i>	the standard USB printer port
<i>/dev/unlpt0</i>	the non-reset USB printer port, use if <i>/dev/ulpt0</i> does not work correctly

netcat (*nc(1)*) can be used to print directly to network printers:

```
% nc nethplaser 9100 < myfile.txt
```

<i>nethplaser</i>	the DNS name of the network printer
<i>9100</i>	the network port used by HP and some other brands

lpd printing using the standard *lpr(1)* is usually more convenient and more versatile than direct printing.

3 Creating A Spool Directory

Printed files are saved temporarily in a spool directory. We'll create one now for use in all of the examples that follow. We'll also set the ownership and permissions to keep print jobs private:

```
# mkdir /var/spool/lpd/lp
# chown daemon:daemon /var/spool/lpd/lp
# chmod 770 /var/spool/lpd/lp
```

4 Enabling *lpd*

The *lpd* server is enabled in */etc/rc.conf*:

```
lpd_enable="YES"
```

On the next startup, *lpd* will be started automatically. Until then, start it manually:

```
# lpd
```

5 Parallel Or USB Port

Create */etc/printcap*:

```
lp:\
    :lp=/dev/lpt0:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/lp:\
    :lf=/var/log/lpd-errs:
```

<i>lp</i>	the standard name for the default print queue
<i>/dev/lpt0</i>	the standard parallel port
<i>/dev/ulpt0</i>	the standard USB printer port
<i>/dev/unlpt0</i>	the non-reset USB printer port, use if <i>/dev/ulpt0</i> does not work correctly
<i>sh</i>	suppress header pages
<i>mx#0</i>	set unlimited file size for print jobs
<i>sd</i>	the spool directory where print jobs will be stored
<i>lf</i>	log file for errors and messages

Tip

The backslashes at the end of those lines are line continuation characters. You could write the whole entry as one long line using colons for separators:

```
lp:lp=/dev/lpt0:sh:mx#0:sd=/var/spool/lpd/lp:lf=/var/log/lpd-errs:
```

Multiple lines are usually easier to read and maintain. Because it's really one long line either way, trying to comment out a section with a *#* will not work.

With this *printcap*, the information sent to the printer is not translated in any way. It's just raw ASCII. You have to make sure that information is in a format that the printer can understand.

Test the printer by sending it some ASCII text:

```
% printf "This is a test\r\n\f" | lpr
```

That's a carriage return, linefeed, and formfeed after the text. Some line printers don't print until an end-of-line character is received, and some page printers don't print until a formfeed is received. So we'll send it all in hopes it will print on whatever sort of freaky printer is on the other end.

If it doesn't print, look your printer up on the net. See if it's called a *Winprinter* or a *host-based* printer, usually along with a lot of cursing. Host-based printers are more work to set up, but often can be made to work on FreeBSD with a *filter*. We'll talk about basic filters in the next section.

6 Adding A Filter

Many Unix applications produce *PostScript* output, but inexpensive printers only understand *PCL*. This filter uses *Ghostscript* to translate PostScript code into PCL. Save it in `/usr/local/libexec/ps2pcl` and then make it executable:

```
#!/bin/sh
/usr/local/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=ljet4 -sOutputFile=- -
```

```
# chmod +x /usr/local/libexec/ps2pcl
```

If Ghostscript is not already on your system, you can install it from the ports collection.

Modify `/etc/printcap` to use the filter:

```
lp:\
    :lp=/dev/lpt0:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/lp:\
    :if=/usr/local/libexec/ps2pcl:\
    :lf=/var/log/lpd-errs:
```

if the input filter, also called a *text filter*

Test the filter by sending a short PostScript program to the printer:

```
% printf "%!\PS\n/Helvetica findfont 24 scalefont setfont \
72 72 moveto (PostScript tested.) show showpage" | lpr
```

This filter expects print jobs to be formatted in PostScript. Many applications already produce PostScript output, like Firefox, OpenOffice.org, and AbiWord. There are also text formatting utilities like *enscript* and image conversion utilities like ImageMagick and GraphicsMagick. All of these are available in the FreeBSD ports collection.

7 Smart Filters

It's nice to have a filter that does the appropriate thing based on the type of data printed. This filter converts text to PostScript using *enscript*, but passes plain PostScript files unfiltered. It's called *psif* as a replacement for the one in the [FreeBSD Handbook](#). Save this file as `/usr/local/libexec/psif`, and make it executable:

```
#!/bin/sh
IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

case "$first_two_chars" in
%!\|033%)
    # %! or ESC% : PostScript job, print it.
    echo "$first_line" && cat && printf "\004" && exit 0
    exit 2
;;
*)
    # otherwise, format with enscript
    ( echo "$first_line"; cat ) \
```

```

    | /usr/local/bin/enscript -o - && printf "\004" && exit 0
    exit 2
    ;;
esac

```

```
# chmod +x /usr/local/libexec/psif
```

This new filter is used in the *if=* property in the *printcap*:

```
lp:\
    :lp=/dev/lpt0:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/lp:\
    :if=/usr/local/libexec/psif:\
    :lf=/var/log/lpd-errs:

```

A couple of popular smart filters available in Ports are *print/apsfilter* and *print/magicfilter*.

8 Network Printers

Set up */etc/printcap* to send jobs to a network printer at the hostname *netlaser*:

```
lp:\
    :lp=\
    :sh:\
    :mx#0:\
    :rm=netlaser:\
    :rp=raw:\
    :sd=/var/spool/lpd/lp:\
    :lf=/var/log/lpd-errs:

```

<i>lp</i>	empty because the printer is not connected to this computer directly
<i>rm</i>	DNS name of the network printer, must be in DNS or <i>/etc/hosts</i>
<i>rp</i>	print queue on the network printer; <i>raw</i> is used by HP printers and others to mean a queue that doesn't do any filtering, just passes data through uncooked. Some print servers use different queue names.

Most network printers support PostScript, so no *if=* filter is needed.

As mentioned earlier, HP and some other printers accept print data at port 9100, and sometimes that works better than using the printer's built-in *lpd* server. That's easily done with an alternate *printcap* entry:

```
lp:\
    :lp=9100@netlaser:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/lp:\
    :lf=/var/log/lpd-errs:

```

9100@netlaser "send data to port 9100 at DNS name 'netlaser'"

9 Different Queues For Different Things

There's no reason to restrict yourself to one queue per printer. You can have a default queue that does some filtering and another for raw, unfiltered throughput, both printing to the same printer.

```
lp:\
    :lp=\
    :sh=\
    :mx#0=\
    :rm=netlaser:\
    :rp=raw:\
    :sd=/var/spool/lpd/lp:\
    :if=/usr/local/libexec/psif:\
    :lf=/var/log/lpd-errs:

rawlaser:\
    :lp=\
    :sh=\
    :mx#0=\
    :rm=netlaser:\
    :rp=raw:\
    :sd=/var/spool/lpd/lp:\
    :lf=/var/log/lpd-errs:
```

Documents printed to the *lp* queue will be filtered through *psif*. Documents printed to the *rawlaser* queue (with `lpr -Prawlaser`) will not be filtered at all.

10 Conclusion

We've just scratched the surface of what is possible with *lpd*. It's a powerful and often-underestimated part of FreeBSD.