4.3BSD Line Printer Spooler Manual

Ralph Campbell

Computer Systems Research Group
Computer Science Division
Department of Electrical Engineering and Computer Science
University of California, Berkeley
Berkeley, CA 94720

ABSTRACT

This document describes the structure and installation procedure for the line printer spooling system developed for the 4.3BSD version of the UNIX* operating system.

Revised June 8, 1993

1. Overview

The line printer system supports:

- multiple printers,
- multiple spooling queues,
- both local and remote printers, and
- printers attached via serial lines that require line initialization such as the baud rate.

Raster output devices such as a Varian or Versatec, and laser printers such as an Imagen, are also supported by the line printer system.

The line printer system consists mainly of the following files and commands:

/etc/printcap	printer configuration and capability data base
/usr/lib/lpd	line printer daemon, does all the real work
/usr/ucb/lpr	program to enter a job in a printer queue
/usr/ucb/lpq	spooling queue examination program
/usr/ucb/lprm	program to delete jobs from a queue
/etc/lpc	program to administer printers and spooling queues
/dev/printer	socket on which lpd listens

The file /etc/printcap is a master data base describing line printers directly attached to a machine and, also, printers accessible across a network. The manual page entry *printcap*(5) provides the authoritative definition of the format of this data base, as well as specifying default values for important items such as the directory in which spooling is performed. This document introduces some of the information that may be placed *printcap*.

---

* UNIX is a trademark of Bell Laboratories.

2. Commands

2.1. lpd – line printer daemon

The program *lpd*(8), usually invoked at boot time from the /etc/rc file, acts as a master server for coordinating and controlling the spooling queues configured in the printcap file. When *lpd* is started it makes a single pass through the *printcap* database restarting any printers that have jobs. In normal operation *lpd* listens for service requests on multiple sockets, one in the UNIX domain (named ‘‘/dev/printer’’) for local requests, and one in the Internet domain (under the ‘‘printer’’ service specification) for requests for printer access from off machine; see *socket* (2) and *services* (5) for more information on sockets and service specifications, respectively. *Lpd* spawns a copy of itself to process the request; the master daemon continues to listen for new requests.

Clients communicate with *lpd* using a simple transaction oriented protocol. Authentication of remote clients is done based on the ‘‘privilege port’’ scheme employed by *rshd* (8C) and *rcmd* (3X). The following table shows the requests understood by *lpd*. In each request the first byte indicates the ‘‘meaning’’ of the request, followed by the name of the printer to which it should be applied. Additional qualifiers may follow, depending on the request.

Request	Interpretation
ˆAprinter\n	check the queue for jobs and print any found
ˆBprinter\n	receive and queue a job from another machine
ˆCprinter [users ...] [jobs ...]\n	return short list of current queue state
ˆDprinter [users ...] [jobs ...]\n	return long list of current queue state
ˆEprinter person [users ...] [jobs ...]\n	remove jobs from a queue

The *lpr* (1) command is used by users to enter a print job in a local queue and to notify the local *lpd* that there are new jobs in the spooling area. *Lpd* either schedules the job to be printed locally, or if printing remotely, attempts to forward the job to the appropriate machine. If the printer cannot be opened or the destination machine is unreachable, the job will remain queued until it is possible to complete the work.

2.2. lpq – show line printer queue

The *lpq* (1) program works recursively backwards displaying the queue of the machine with the printer and then the queue(s) of the machine(s) that lead to it. *Lpq* has two forms of output: in the default, short, format it gives a single line of output per queued job; in the long format it shows the list of files, and their sizes, that comprise a job.

2.3. lprm – remove jobs from a queue

The *lprm* (1) command deletes jobs from a spooling queue. If necessary, *lprm* will first kill off a running daemon that is servicing the queue and restart it after the required files are removed. When removing jobs destined for a remote printer, *lprm* acts similarly to *lpq* except it first checks locally for jobs to remove and then tries to remove files in queues off-machine.

2.4. lpc – line printer control program

The *lpc*(8) program is used by the system administrator to control the operation of the line printer system. For each line printer configured in /etc/printcap, *lpc* may be used to:

- disable or enable a printer,
- disable or enable a printer’s spooling queue,
- rearrange the order of jobs in a spooling queue,
- find the status of printers, and their associated spooling queues and printer daemons.

3. Access control

The printer system maintains protected spooling areas so that users cannot circumvent printer accounting or remove files other than their own. The strategy used to maintain protected spooling areas is as follows:

- The spooling area is writable only by a *daemon* user and *daemon* group.
- The *lpr* program runs set-user-id to *root* and set-group-id to group *daemon*. The *root* access permits reading any file required. Accessibility is verified with an *access* (2) call. The group ID is used in setting up proper ownership of files in the spooling area for *lprm*.
- Control files in a spooling area are made with *daemon* ownership and group ownership *daemon*. Their mode is 0660. This insures control files are not modified by a user and that no user can remove files except through *lprm*.
- The spooling programs, *lpd*, *lpq*, and *lprm* run set-user-id to *root* and set-group-id to group *daemon* to access spool files and printers.
- The printer server, *lpd*, uses the same verification procedures as *rshd* (8C) in authenticating remote clients. The host on which a client resides must be present in the file /etc/hosts.equiv or /etc/hosts.lpd and the request message must come from a reserved port number.

In practice, none of *lpd*, *lpq*, or *lprm* would have to run as user *root* if remote spooling were not supported. In previous incarnations of the printer system *lpd* ran set-user-id to *daemon*, set-group-id to group *spooling*, and *lpq* and *lprm* ran set-group-id to group *spooling*.

4. Setting up

The 4.3BSD release comes with the necessary programs installed and with the default line printer queue created. If the system must be modified, the makefile in the directory /usr/src/usr.lib/lpr should be used in recompiling and reinstalling the necessary programs.

The real work in setting up is to create the *printcap* file and any printer filters for printers not supported in the distribution system.

4.1. Creating a printcap file

The *printcap* database contains one or more entries per printer. A printer should have a separate spooling directory; otherwise, jobs will be printed on different printers depending on which printer daemon starts first. This section describes how to create entries for printers that do not conform to the default printer description (an LP-11 style interface to a standard, band printer).

4.1.1. Printers on serial lines

When a printer is connected via a serial communication line it must have the proper baud rate and terminal modes set. The following example is for a DecWriter III printer connected locally via a 1200 baud serial line.

```
lp|LA-180 DecWriter III:\
        :lp=/dev/lp:br#1200:fs#06320:\
        :tr=\f:of=/usr/lib/lpf:lf=/usr/adm/lpd-errs:
```

The **lp** entry specifies the file name to open for output. Here it could be left out since ‘‘/dev/lp’’ is the default. The **br** entry sets the baud rate for the tty line and the **fs** entry sets CRMOD, no parity, and XTABS (see *tty* (4)). The **tr** entry indicates that a form-feed should be printed when the queue empties so the paper can be torn off without turning the printer off-line and pressing form feed. The **of** entry specifies the filter program *lpf* should be used for printing the files; more will be said about filters later. The last entry causes errors to be written to the file ‘‘/usr/adm/lpd-errs’’ instead of the console. Most errors from *lpd* are logged using *syslogd* (8) and will not be logged in the specified file. The filters should use *syslogd* to report errors; only those that write to standard error output will end up with errors in the **lf** file. (Occasionally errors sent to standard error output have not appeared in the log file; the use of *syslogd* is highly recommended.)

4.1.2. Remote printers

Printers that reside on remote hosts should have an empty **lp** entry. For example, the following printcap entry would send output to the printer named ‘‘lp’’ on the machine ‘‘ucbvax’’.

```
lp|default line printer:\
        :lp=:rm=ucbvax:rp=lp:sd=/usr/spool/vaxlpd:
```

The **rm** entry is the name of the remote machine to connect to; this name must be a known host name for a machine on the network. The **rp** capability indicates the name of the printer on the remote machine is ‘‘lp’’; here it could be left out since this is the default value. The **sd** entry specifies ‘‘/usr/spool/vaxlpd’’ as the spooling directory instead of the default value of ‘‘/usr/spool/lpd’’.

4.2. Output filters

Filters are used to handle device dependencies and to do accounting functions. The output filtering of **of** is used when accounting is not being done or when all text data must be passed through a filter. It is not intended to do accounting since it is started only once, all text files are filtered through it, and no provision is made for passing owners’ login name, identifying the beginning and ending of jobs, etc. The other filters (if specified) are started for each file printed and do accounting if there is an **af** entry. If entries for both **of** and other filters are specified, the output filter is used only to print the banner page; it is then stopped to allow other filters access to the printer. An example of a printer that requires output filters is the Benson-Varian.

```
va|varian|Benson-Varian:\
        :lp=/dev/va0:sd=/usr/spool/vad:of=/usr/lib/vpf:\
        :tf=/usr/lib/rvcat:mx#2000:pl#58:px=2112:py=1700:tr=\f:
```

The **tf** entry specifies ‘‘/usr/lib/rvcat’’ as the filter to be used in printing *troff*(1) output. This filter is needed to set the device into print mode for text, and plot mode for printing *troff* files and raster images (see *va*(4V)). Note that the page length is set to 58 lines by the **pl** entry for 8.5" by 11" fan-fold paper. To enable accounting, the varian entry would be augmented with an **af** filter as shown below.

```
va|varian|Benson-Varian:\
        :lp=/dev/va0:sd=/usr/spool/vad:of=/usr/lib/vpf:\
        :if=/usr/lib/vpf:tf=/usr/lib/rvcat:af=/usr/adm/vaacct:\
        :mx#2000:pl#58:px=2112:py=1700:tr=\f:
```

4.3. Access Control

Local access to printer queues is controlled with the **rg** printcap entry.

```
        :rg=lprgroup:
```

Users must be in the group *lprgroup* to submit jobs to the specified printer. The default is to allow all users access. Note that once the files are in the local queue, they can be printed locally or forwarded to another host depending on the configuration.

Remote access is controlled by listing the hosts in either the file /etc/hosts.equiv or /etc/hosts.lpd, one host per line. Note that *rsh*(1) and *rlogin*(1) use /etc/hosts.equiv to determine which hosts are equivalent for allowing logins without passwords. The file /etc/hosts.lpd is only used to control which hosts have line printer access. Remote access can be further restricted to only allow remote users with accounts on the local host to print jobs by using the **rs** printcap entry.

```
        :rs:
```

5. Output filter specifications

The filters supplied with 4.3BSD handle printing and accounting for most common line printers, the Benson-Varian, the wide (36") and narrow (11") Versatec printer/plotters. For other devices or accounting methods, it may be necessary to create a new filter.

Filters are spawned by *lpd* with their standard input the data to be printed, and standard output the printer. The standard error is attached to the **lf** file for logging errors or *syslogd* may be used for logging errors. A filter must return a 0 exit code if there were no errors, 1 if the job should be reprinted, and 2 if the job should be thrown away. When *lprm* sends a kill signal to the *lpd* process controlling printing, it sends a SIGINT signal to all filters and descendents of filters. This signal can be trapped by filters that need to do cleanup operations such as deleting temporary files.

Arguments passed to a filter depend on its type. The **of** filter is called with the following arguments.

filter –**w**width –**l**length

The *width* and *length* values come from the **pw** and **pl** entries in the printcap database. The **if** filter is passed the following parameters.

filter [–**c**] –**w**width –**l**length –**i**indent –**n** login –**h** host accounting_file

The –**c** flag is optional, and only supplied when control characters are to be passed uninterpreted to the printer (when using the –**l** option of *lpr* to print the file). The –**w** and –**l** parameters are the same as for the **of** filter. The –**n** and –**h** parameters specify the login name and host name of the job owner. The last argument is the name of the accounting file from *printcap*.

All other filters are called with the following arguments:

filter –**x**width –**y**length –**n** login –**h** host accounting_file

The –**x** and –**y** options specify the horizontal and vertical page size in pixels (from the **px** and **py** entries in the printcap file). The rest of the arguments are the same as for the **if** filter.

6. Line printer Administration

The *lpc* program provides local control over line printer activity. The major commands and their intended use will be described. The command format and remaining commands are described in *lpc*(8).

abort and **start**

Abort terminates an active spooling daemon on the local host immediately and then disables printing (preventing new daemons from being started by *lpr*). This is normally used to forcibly restart a hung line printer daemon (i.e., *lpq* reports that there is a daemon present but nothing is happening). It does not remove any jobs from the queue (use the *lprm* command instead). *Start* enables printing and requests *lpd* to start printing jobs.

enable and **disable**

Enable and *disable* allow spooling in the local queue to be turned on/off. This will allow/prevent *lpr* from putting new jobs in the spool queue. It is frequently convenient to turn spooling off while testing new line printer filters since the *root* user can still use *lpr* to put jobs in the queue but no one else can. The other main use is to prevent users from putting jobs in the queue when the printer is expected to be unavailable for a long time.

restart

Restart allows ordinary users to restart printer daemons when *lpq* reports that there is no daemon present.

stop

Stop halts a spooling daemon after the current job completes; this also disables printing. This is a clean way to shutdown a printer to do maintenance, etc. Note that users can still enter jobs in a spool queue while a printer is *stopped*.

topq

Topq places jobs at the top of a printer queue. This can be used to reorder high priority jobs since *lpr* only provides first-come-first-serve ordering of jobs.

7. Troubleshooting

There are several messages that may be generated by the the line printer system. This section categorizes the most common and explains the cause for their generation. Where the message implies a failure, directions are given to remedy the problem.

In the examples below, the name *printer* is the name of the printer from the *printcap* database.

7.1. LPR

lpr: *printer* **: unknown printer**

The *printer* was not found in the *printcap* database. Usually this is a typing mistake; however, it may indicate a missing or incorrect entry in the /etc/printcap file.

lpr: *printer* **: jobs queued, but cannot start daemon.**

The connection to *lpd* on the local machine failed. This usually means the printer server started at boot time has died or is hung. Check the local socket /dev/printer to be sure it still exists (if it does not exist, there is no *lpd* process running). Usually it is enough to get a super-user to type the following to restart *lpd*.

```
% /usr/lib/lpd
```

You can also check the state of the master printer daemon with the following.

```
% ps l‘cat /usr/spool/lpd.lock‘
```

Another possibility is that the *lpr* program is not set-user-id to *root*, set-group-id to group *daemon*. This can be checked with

```
% ls –lg /usr/ucb/lpr
```

lpr: *printer* **: printer queue is disabled**

This means the queue was turned off with

% lpc disable *printer*

to prevent *lpr* from putting files in the queue. This is normally done by the system manager when a printer is going to be down for a long time. The printer can be turned back on by a super-user with *lpc*.

7.2. LPQ

waiting for *printer* **to become ready (offline ?)**

The printer device could not be opened by the daemon. This can happen for several reasons, the most common is that the printer is turned off-line. This message can also be generated if the printer is out of paper, the paper is jammed, etc. The actual reason is dependent on the meaning of error codes returned by system device driver. Not all printers supply enough information to distinguish when a printer is off-line or having trouble (e.g. a printer connected through a serial line). Another possible cause of this message is some other process, such as an output filter, has an exclusive open on the device. Your only recourse here is to kill off the offending program(s) and restart the printer with *lpc*.

printer **is ready and printing**

The *lpq* program checks to see if a daemon process exists for *printer* and prints the file *status* located in the spooling directory. If the daemon is hung, a super user can use *lpc* to abort the current daemon and start a new one.

waiting for *host* to come up

This implies there is a daemon trying to connect to the remote machine named *host* to send the files in the local queue. If the remote machine is up, *lpd* on the remote machine is probably dead or hung and should be restarted as mentioned for *lpr*.

sending to *host*

The files should be in the process of being transferred to the remote *host*. If not, the local daemon should be aborted and started with *lpc*.

Warning: *printer* is down

The printer has been marked as being unavailable with *lpc*.

Warning: no daemon present

The *lpd* process overseeing the spooling queue, as specified in the ‘‘lock’’ file in that directory, does not exist. This normally occurs only when the daemon has unexpectedly died. The error log file for the printer and the *syslogd* logs should be checked for a diagnostic from the deceased process. To restart an *lpd*, use

% lpc restart *printer*

no space on remote; waiting for queue to drain

This implies that there is insufficient disk space on the remote. If the file is large enough, there will never be enough space on the remote (even after the queue on the remote is empty). The solution here is to move the spooling queue or make more free space on the remote.

7.3. LPRM

lprm: *printer* : cannot restart printer daemon

This case is the same as when *lpr* prints that the daemon cannot be started.

7.4. LPD

The *lpd* program can log many different messages using *syslogd* (8). Most of these messages are about files that can not be opened and usually imply that the *printcap* file or the protection modes of the files are incorrect. Files may also be inaccessible if people manually manipulate the line printer system (i.e. they bypass the *lpr* program).

In addition to messages generated by *lpd*, any of the filters that *lpd* spawns may log messages using *syslogd* or to the error log file (the file specified in the **lf** entry in *printcap*).

7.5. LPC

couldn’t start printer

This case is the same as when *lpr* reports that the daemon cannot be started.

cannot examine spool directory

Error messages beginning with ‘‘cannot ...’’ are usually because of incorrect ownership or protection mode of the lock file, spooling directory or the *lpc* program.