LPRng Tutorial

Patrick Powell" "<papowell@sdsu.edu>"

ABSTRACT

This is a tutorial on the installation and configuration of the LPRng software. The various *man* pages provide details on each of the LPRng components and should be consulted while proceeding through the tutorial. The following topics are covered: setting up and testing a printcap file; setting up and testing authorization; accounting; setting up network databases; debugging problems.

1. Introduction

This tutorial is an introduction to the installation and testing of the LPRng software. It assumes that you are an oridinary user, and that you have source to the LPRng code. Part of the LPRng distribution is a *test* configuration that allows you to perform all of the tutorial exercises as an ordinary user. The following topics will be covered.

1. Setting up the TESTSUPPORT tutorial system.
2. Basic Management Functions using LPC
3. Creating a printcap file for the LPRng client programs such as **lpr**, **lpq**, and **lpc**.
4. Server (printer) printcap information and spool queues.
5. Permissions and authorizations.
6. Setting up a general permissions file.
7. Setting up a per printer permissions file.
8. Basic accounting.

2. TESTSUPPORT

A simple set of test and support files is provided in the `TESTSUPPORT` directory of the LPRng distribution. These files, when processed and installed in the `/tmp` directory, will create a simple user level printer spooler system.

```
LPD       - copy of spool directories
Makefile  - generation and installation
clearlog  - shell script to clear log files
dbserver.sh   - prototype database server
filter.sh - prototype shell based filter
lpd.conf.proto   - prototype configuration file
common.conf.proto   - prototype configuration file
lpd.perms.proto  - prototype permissions file
main.c    - prototype C based filter
makedb    - database creation script
printcap.proto   - prototype printcap file
```

To install the files, carry out the following procedure in the `TESTSUPPORT` directory.

```
CSH -
    setenv HOST fully qualified host name
    setenv USER userid
    setenv LPD_CONF /tmp/LPD/lpd.conf
    path=( /tmp/LPD $path )

Bourne Shell
    HOST=fully qualified host name
    USER=userid
    LPD_CONF=/tmp/LPD/lpd.conf
    export HOST USER LPD_CONF
    PATH=/tmp/LPD:$PATH

make
```

To check that the installation was successful, the command `lpd -D2 -F` should produce output similar to the following.

```
papowell@taco:70>lpd -D2 -F
Sep 17 16:13:17 lpd [3665]  *** Get_debug_parms ***
Sep 17 16:13:17 lpd [3665]  debug: 2
Sep 17 16:13:17 lpd [3665]  test: 0
Sep 17 16:13:17 lpd [3665]  remote: 0
Sep 17 16:13:17 lpd [3665]  log: <NULL>
Sep 17 16:13:17 lpd [3665]  *** <END> ***
Sep 17 16:13:17 lpd [3665]  main: Configuration file '/tmp/LPD/lpd.conf'
Sep 17 16:13:17 lpd [3665]  seteuid_wrapper: Before RUID/EUID 118/118,
Sep 17 16:13:17 lpd [3665]  seteuid_wrapper: After uid/euid 118/118
Sep 17 16:13:17 taco lpd [3665]  Getprintcap: paths '/tmp/LPD/printcap....
Sep 17 16:13:17 taco lpd [3665]  Printcap file '/tmp/LPD/printcap....
Sep 17 16:13:17 taco lpd [3665]  Get_perms: /tmp/LPD/lpd.perms.taco.sdsu.edu
Sep 17 16:13:17 taco lpd [3665]  Permissions file '/tmp/LPD/lpd....
Sep 17 16:13:17 taco Do_queue_jobs [3667]  t1: Open_log: log file log
Sep 17 16:13:18 taco Do_queue_jobs [3668]  t2: Open_log: log file log
Sep 17 16:13:19 taco Do_queue_jobs [3669]  t4: Open_log: log file log
...
```

3. LPQ

The `lpq` program displays the current status of one or more spool queues.

1. In one window, start the LPD server using `lpd -F`.
2. In another window, run the `lpq` program. You should see a status display similar to the following:

```
Printer: t1@taco 'Test Printer 1'
  Queue: 9 printable jobs
  Warning: 1 held jobs in queue
  Server: pid 4537 active, Unspooler: pid 4538 active
  Status: printing 'cfA637taco', sending FF after banner at 18:29:54
  Status: printing job 'cfA637taco data file 'dfA637taco', format 'f' at 18:29:54
  Status: printed 0% of 15 bytes at 18:29:54
  Status: printed all 15 bytes at 18:29:54
 Rank  Owner           Class Job Files                           Size Time
active papowell@taco       A 637 VERSION                           15 18:27:28
2      papowell@taco       A 004 VERSION                           15 18:27:27
3      papowell@taco       A 007 (stdin)                            5 18:27:27
4      papowell@taco       A 067 VERSION                           15 18:27:27
5      papowell@taco       A 070 VERSION                           15 18:27:27
6      papowell@taco       A 343 VERSION                           16 18:27:28
7      papowell@taco       A 672 VERSION                           15 18:27:28
8      papowell@taco       A 692 VERSION                           15 18:27:28
9      papowell@taco       A 336 VERSION                           15 18:29:55
hold   papowell@taco       A 335 VERSION                           15 18:27:27
error  papowell@taco       A 002 ERROR: cannot open '/tmp/LPD/t1/dfA003taco'
error  papowell@taco       A 003 ERROR: cannot open '/tmp/LPD/t1/dfA003taco'
error  ???@???             A 001 ERROR: empty control file 2 mins old
```

Try running `lpq -a`; this will display the status of all spool queues. The command `lpq -t3 -c` will display the status, sleep for 3 seconds, and then clear the screen before redisplaying status.

4. LPR and LPRM

The `lpr` program spools files to the output. Try the following commands:

```
echo hi | lpr
lpq
```

The status display will show your spooled job. Now try spooling several jobs:

```
echo hi | lpr
echo hi | lpr
echo hi | lpr
echo hi | lpr
lpq
```

You can remove a spooled job by using: `lprm` *jobnumber*. Try removing one of the jobs in the spool queue. If you want to remove all jobs, just use `lprm` without a job number; by default it removes all of your jobs.

5. LPC

The `lpc` program displays the current status of one or more spool queues.

1. In one window, start the LPD server using `lpd -F`.
2. The `lpc` program can run in interactive or line mode. We will first run it in interactive mode. In another window, run the `lpc` program. You should see a status display similar to the following:

   ```
   lpc >
   ```

   Try entering the help command: `lpc > help`; this will display the help information.

```
usage: lpc [-Ddebuglevel] [-V] [-Pprinter] [commands]
 with no commands, reads from stdin  -Pprinter    - specify printer
  -V          - increase information verbosity
  -Ddebuglevel - debug level
 commands:
 abort   (printer[@host] | all)  - stop server
 autohold (printer[@host] | all)  - autohold on
 disable (printer[@host] | all)  - disable queueing
 debug   (printer[@host] | all) debugparms - set debug level for printer
 enable  (printer[@host] | all)  - enable  queueing
 hold    (printer[@host] | all) (name[@host] | job | all)* - hold job
 kill    (printer[@host] | all)  - stop and restart server
 lpd [HUP]  - get LPD PID, signal it to reread printcap and configuration
 lpq (printer[@host] | all) (name[@host] | job | all)*     - invoke LPQ
 lprm (printer[@host] | all) (name[@host]|host|job| all)*  - invoke LPRM
 noautohold (printer[@host] | all)  - autohold off
 printcap (printer[@host] | all) - report printcap values
 quit                            - exit LPC
 redirect (printer[@host] | all) (printer@host | off )*    - redirect jobs
 release  (printer[@host] | all) (name[@host] | job | all)* - release job
 reread                          - LPD reread database information
 start   (printer[@host] | all)  - start printing
 status  (printer[@host] | all)  - status of printers
 stop    (printer[@host] | all)  - stop  printing
 topq    (printer[@host] | all) (name[@host] | job | all)* - reorder job
```

The `lpc` command can be used to start and stop printing, spooling, and redirect jobs to other queues. By default, most commands work on the currently selected printer, which is the users default printer or the last selected printer.

5.1. Status Reports

The `status` command can be used to show the status of printers.

```
lpc >status all
Printer              Printing Spooling Jobs  Server   Slave Redirect Debug
t1@taco               enabled  enabled    3    none    none
t2@taco               enabled  enabled    0    none    none
t3@taco               enabled  enabled    0    none    none
t4@taco               enabled  enabled    0    none    none
t5@taco               enabled  enabled    0    none    none
t6@taco               enabled  enabled    0    none    none
t3@taco               enabled  enabled    0    none    none
```

The status display shows the name of the printer, its print and spooling status, the number of jobs in the queue, if there is a server process active, and if there is a slave process for the server process. The redirect field indicates if the queue is being redirected to another print queue and the debug field is the current debugging level set by the lpc command.

5.2. Starting and Stopping Printing and Spooling

The *start* and *stop* commands can be used to start and stop printing.

1. Enter the command `stop t1`, `status`, `start`, and `status` as shown below.

```
lpc >stop t1
t1 stopped
lpc >status
Printer              Printing Spooling Jobs  Server   Slave Redirect Debug
t1@taco              disabled  enabled    3    none    none
lpc >start
t1 started
lpc >status
Printer              Printing Spooling Jobs  Server   Slave Redirect Debug
t1@taco               enabled  enabled    3    none    none
```

The `enable` and `disable` commands can be used to enable or disable spooling to the queue in a similar manner.

```
lpc >disable t1
t1 disabled
lpc >status
Printer              Printing Spooling Jobs  Server   Slave Redirect Debug
t1@taco               enabled disabled    3    none    none
lpc >enable t1
t1 enabled
lpc >status
Printer              Printing Spooling Jobs  Server   Slave Redirect Debug
t1@taco               enabled  enabled    3    none    none
```

5.3. LPQ and LPRM Commands

For convenience, the `lprm` and `lpq` commands are available in interactive mode. This allows jobs to be removed from a print queue and the status of the print queue to be viewed after performing commands.

5.4. Killing and Aborting Print Queue

The abort and kill commands are used to kill of printing of a job. The kill command will perform an abort operation followed by a start operation.

The abort command is useful when a job is being printed and needs to be restarted, but at a later time using the start command. The kill command is useful when a job can be immediately restarted.

5.5. TOPQ - Changing job priority

The topq command can be used to reorder the priority of user jobs. Jobs that have their priority altered will be printed before jobs that have not. Updated jobs will be printed in last updated, first out order.

For example, the command `topq root` will have all of the root users jobs printed first, starting with the oldest job in the queue.

5.6. Hold, Autohold, and Release

The hold, autohold, noautohold, and release commands are used to temporarily prevent a job from being printed. The hold command is used to hold a job and the release command to release it.

The autohold option can be enabled in the `printcap` file using the `:ah:` printcap field. If autohold is enabled, then all jobs will be held until released using the release command. The interactive autohold command can be used to override the printcap information. The status of the autohold option is recorded in the spool queue `control` file and will persist over system restarts. Autohold can be turned off using the noautohold command.

When a job is released the printer may need to be started if it is disabled.

5.7. Job and Queue Redirection

The move and redirect commands are used to move jobs between different queues.

The move command will move a selected set of jobs to a new queue. For example, `move t1 root t2` will move all of the jobs on printer t1 submitted by root to the t2 printer.

The redirect command will cause all jobs submitted to a particular spool queue to be forwarded to the new printer. The `redirect off` command will turn redirection off.

5.8. Diagnostic Commands - debug, lpd, reread, printcap

The debug, lpd, and lpq commands are available for diagnostic purposes.

The debug command allows the adminstrator to set debugging information and values for a specific spool queue. For example, `debug 7,remote=5,log=/tmp/x` would set debugging to level 7, the remote flag to 5, and send debugging log information to the `/tmp/x` file. The `debug` command with no options will turn all debugging options off.

The lpd command will print the process ID (PID) of the LPD server process. The reread command will send it a `SIGHUP` signal, causing it to reread the various databases.

The printcap command will cause the printcap entry for the specified printer to be returned. This is useful when trying to determine the actual values of a printcap entry being used by the LPRng software.

6. Command Line Interface

The command line interface is identical to the interactive interface, but only allows a single command to be entered at a time. For example: `lpc status all` will print the status of all the printers, and `lpc start all` will start all printers.

7. LPRM

The `lprm` program is used to remove jobs from the spool queues. For the following exercises, we will temporarily stop printing, and submit jobs.

1. In one window, start the LPD server using `lpd -F`.
2. Temporarily stop printing for all spool queues.

   ```
   lpc stop all
   ```

3. Submit a set of jobs to a printer.

   ```
   echo hi |lpr -Pt1
   echo hi |lpr -Pt1
   ```

4. Display the queue status using `lpq -a`.

   ```
   Printer: t1@taco 'Test Printer 1' (printing disabled)
     Queue: 5 printable jobs
     Warning: no server present
     Status: finished operations at Sun Mar  3 13:51:49 1996
    Rank  Owner           Class Job Files                                 Size Time
   1      papowell@taco       A 050 (stdin)                                  3 13:53:39
   2      papowell@taco       A 068 (stdin)                                  3 13:55:41
   3      papowell@taco       A 075 (stdin)                                  3 13:55:46
   ```

5. Try removing one of the jobs using `lprm` *jobnumber*; for example `lprm 050`.
6. Try using `lprm` *username* to remove all jobs submitted by a particular user.
7. If you want to remove all jobs in a queue and you have control permissions on the queue, you can use `lprm all`.

8. The `-a` (all) option will try to remove the specified jobs from all printers. Try `lprm -a all` to remove all jobs for all users.

8. Printcaps and CHECKPC

The `checkpc` program can be used to validate printcap entries and to set permissions and ownerships on the various printcap files. The following is a set of exercises intended to show the use of the checkpc program.

9. Stop the lpd server by sending it a kill signal. A quick method of doing this is:

```
kill `ps -aux |grep lpd | awk '{print $2 }'`
```

10. Run the `checkpc` program. The following is typical output.

```
 LPRng version LPRng-2.2.0
 Get_perms: permissions file '/tmp/LPD/lpd.perms.taco.sdsu.edu'
 Printcap file '/tmp/LPD/printcap.taco.sdsu.edu'
 Printcap file '/tmp/LPD/lpd_printcap.taco.sdsu.edu'
 LPD lockfile '/tmp/LPD/lpd.lock.taco.sdsu.edu'
   checking file '/tmp/LPD/lpd.lock.taco.sdsu.edu'
 t1: Checking printcap entry 't1'
 t1:   checking file '/tmp/LPD/t1/control.t1'
 t1:   checking file '/tmp/LPD/t1/status.t1'
 t1:   checking file '/tmp/LPD/t1/log'
 t1:   checking file '/tmp/LPD/t1/accnt'
checkpc: Warning - permissions of '/tmp/LPD/t1/cfA001taco' are 0644, not 0600
 t1: Checking log file '/tmp/LPD/t1/log'
 t1:    log file 12060 bytes long: no truncation
```

11. Now try using the `-f` (fix) option to set permissions and ownership.

```
checkpc -f
```

Permissions and ownership will be corrected.

12. As the lpd server executes, it will put output into log, status, and accounting files in the spool directory. The `checkpc -t` *size* command will truncate these files to *size* bytes. For example, `checkpc -t 2k` truncate these files to less than 2K bytes. Use this command to truncate the log files.

9. lpd.perms File and Permissions

The lpd server uses the `lpd.perms` file to control user access to services. The following is an example permissions file.

```
# Sample permissions file
DEFAULT ACCEPT
# Root on server can do control operations
ACCEPT SERVICE=C SERVER USER=root
# The specified user on the server can do control operations
ACCEPT SERVICE=C SERVER USER=user
# Do not allow any other users to do control operations
REJECT SERVICE=C
# Job Removal can be done by the same user on the originating host
ACCEPT SERVICE=M SAMEHOST SAMEUSER
# Also by root on the server
ACCEPT SERVICE=M SERVER USER=root
# Reject all other requests
REJECT SERVICE=M
# Reject requests from subnet 130.191.25.0 (24 bit Subnet mask)
REJECT IP=130.191.25.0/255.255.255.0
# or REJECT IP=130.191.25.0/24
# Printer t1: Accept all requests
PRINTER=t1 ACCEPT
```

The man page for `lpd.perms.` has a detailed description of the various fields and entries. Each line in the permissions file is a test that is performed against various information associated with a user request. The tests are applied in sequence until a match is found; the `ACCEPT` or `REJECT` keyword specifies of the request is accepted or rejected.

The `DEFAULT` keyword specifies the default action if no explicit match is found.

In addition to the main permissions file, each spool queue can have an additional set of permissions tests that can be applied to requests that have been accepted by the main permissions file.

The `SERVICE` tag specifies the type of service for the match. Note that not all service requests have the same amount of information to be tested. This is a consequence of the protocol used to send requests to the server. In addition, the `X` (connection) request is applied to all connections, and can be used to restrict requests to only a limited range of host systems.

When constructing a permissions file, the most specific tests should be applied first, then the most general. The `REJECT` tests should precede the `ACCEPT`, and a default for a particular action should immediately follow. Here are a suggested set of exerises for the reader.

13. Modify the permissions file so that print requests and job from `user` are rejected. Answer:

```
REJECT SERVICE=R,P USER=user
```

14. Modify the permissions file to allow only hosts in the IP address submask 130.191.130.0/255.255.255.0 to have access. Answer:

```
REJECT SERVICE=X NOT IP=130.191.130.0/255.255.255
```