



## Linux Printing

### Introduction and/or Background

Printing is one of the most confusing topics in discussions about Unix/Linux. Print is one of the oldest functions in the OS, hence a long history that has had updates and additions over time. In addition some commands are implemented differently. In AT&T System V the `lpr` command is implemented as a device (e.g. `/dev/lp`) while in many Linux systems `lpr` is treated as a kernel subsystem. Finally as a unifying effort an overarching printing superstructure was developed – CUPS, which will be treated in a different lab.

Focus on the process to define some clarity. Afterwards we will cover individual commands.

Text -> [Formatter] -> Spooler -> Print Driver -> Print Device

Other than the Formatter the other steps are followed for every job submitted for print. The Text is generally sourced from files, but can also be provided by STDIN via the keyboard. The Text is sent to the Spooler. Spooler accepts incoming jobs, assigns a job number and places the job in the print queue. The queue is a first-in, first-out stack unless the device itself supports print priority as a selection criteria. The Spooler delivers the next in line job to the Print Driver of the selected Print Device. Recognize two facts - The Spooler already knows which Print Device the job is to be delivered and the proper Print Driver to utilize. Print Drivers are typically formatted as a `.ppd` file and are supplied by the printer manufacturer. See `/usr/share/ppd` as examples. Finally, the job is routed. Routed, because these days Print Devices as are likely to be networked via Ethernet or wireless technology as being locally attached.

In a more advanced state of affairs the Formatter can be utilized to pull all sorts of printing tricks. Here are some examples:

- Print to a fax card or network fax delivery system. The Text is mangled into a `.tiff` fax format and routed by a user provided POTS number to the destination fax address.
- Next available. The job is delivered to the same printer type that indicates a ready status. This is quite common in large offices where there are dedicated print rooms to reduce noise and hold a gang of printers all of the same mfr/type.
- Gang print. Need a thousand copies of a 10 page document? Submit the Text to X

number of printers of the same class. Class being a Print Device that can use the same .pdd file. A large volume of print can be accomplished using moderate priced devices. (e.g. a print version of RAID.)

Confused? Well consider in a networked environment the Spooler -> Print Driver -> Print Device stream can be on a computer from the source of the Text. Quite common in large environments that have different devices, job accounting and control of who may access what device types.

### *Commands*

Of concern to the operator are four components - Text, Formatter, Spooler, Print Driver

### *Text*

Text can be delivered various ways:

- `cat "some file" > /dev/lp` under the old AT&T System V compatible systems.
- `cat "some file" | lpr` in many Linux systems
- Any application that supports a print subroutine. E.g. Nano, Vim, Libre Office.

Without a directive of where, the output goes to the default printer device for the system of origin.

### *Formatter*

The problem in a Unix environment is that Text is generally formatted for the default screen format, 40, 80, 132 chars across. The Print Device however may not support that choice. This is where the Formatter comes in. It comes in and rearranges the Text into a different format supported by the Print Device.

- `pr`, which permits the operator to specify document page length, columns across, header, footer, etc.
- `fmt`, has all the features of `pr`, with additions, preserve first line indentation, uniform spacing, split long lines, etc. It was originally designed to support the mail function.

### *Spooler*

- `lp`, `lpr` - `lp` is short for line printer for back in the days when bar code line print was the only game in town. `lpr` is a revision of `lp` that also can direct output to networked devices. `lpr` is the dominant command line tool for printing. Example:

```
cat myfile.txt | lpr
lpr -Plp evtmlib.c
```

- `lpq` - Displays the print queue that the Spooler is holding. Lacking anything else the

content displayed is the default the Print Device. Options provide for other devices or remote print queues. Example:

```
lpq -P lp0
```

lp0 is the destination device of all jobs in the queue that match.

```
drdog@drdog:/var/log$ lpq -P NullPrinter
NullPrinter is ready
Rank  Owner  Job  File(s)  Total Size
1st   drdog   6    dmesg    80896 bytes
2nd   drdog   7    dmesg    80896 bytes
```

- **lprm** - Removes a print job from the queue based on the job number assigned by the Spooler. Depending on user rights jobs may be displayed for which they are not the owner and cannot delete. Normal users generally only see their pending jobs in the queue. Example:

```
lprm 485
```

- **lpadmin** - This command is part of CUPS but is provided here for completeness. lpadmin provides the same functions as lpc (see below) for CUPS designated print devices. Example:

```
lpadmin -p LaserJet -E -v socket://10.1.1.1 -m laserjet.ppd
```

- **lpc** - Controls the status of a particular Print Device. lpc can start, stop, reassign queues that are headed for a particular destination. Generally in a shared server, the lpc command is only accessible to the root user after OS installation. Example:

```
lpc status
```

### *Print Driver*

Or where is my print definition (.ppd) file? If I don't have it, where can I get it? There are four sources.

### *Device Mfr*

First place to look is the company that made your printer. Their support website will typically have a copy of the .ppd file for you to download. They will also generally have the latest update that is newer than what your distro distribution will have on hand for the most common printer types.

### *Foomatic*

Foomatic is a database driven system that maintains files of nearly all the printers out there. Interesting twist with Foomatic, their drivers are direct, and don't use the spooler. That means most of your selections must be a locally attached device. Foomatic is now part of the next choice -

### *OpenPrint*

OpenPrint is part of the IEEE working group on print device connectivity, including drivers. OpenPrint would be the second best choice for a driver after the mfr of the device. Most of the drivers provide network support as well.

### *PDF*

As a last resort print your document to a PDF file, then send the file to the printer. The majority of top selling printers have the PDF language pack embedded into the hardware permitting it to interpret a PDF natively. The downside is you are limited in the fonts available to the PDF driver embedded in the printer.

## **Objectives**

In this project/lab the student will:

- Gain familiarity with Linux printing

## **Equipment/Supplies Needed**

- As specified in Lab 0.0.1.

## **Procedure**

Perform the steps in this lab in the order they are presented to you. Answer all questions and record the requested information. Use the Linux Virtual Machine to perform lab activities as directed. Unless otherwise stated, all tasks done as a non-root user. If root access is needed use the sudo command.

## **Assignment**

### *Setup printing*

First confirm that the following commands are available by checking their man pages:

1. `man lpr`
2. `man lpq`
3. `man lprm`
4. `man lpstat`
5. `man pr`

6. `man fmt`
7. `man lpadmin`

If you get a 'not found' response it will be necessary to install that command from the repositories.

Next, rather than burning a lot of paper or if you do not have a printer, you need to set up a null printer. Bring up the printer manager in your distribution and then follow the steps in the video 'NullPrinter'.

lpr, lpq, lprm, lpstat are print control commands to submit, manage or remove print jobs. pr, fmt are print text formatters. Finally lpadmin is the command line tool for CUPS, a topic we will get into later.

### *Formatting*

Promote the user to root user account. Go to the following directory -

9. `cd /var/log`

Display the file 'dmesg'

10. `dmesg > smesg && cat smesg`

Now try this:

11. `pr -w 60 smesg`

Can you describe what the difference is between the two outputs? Open a Word or Writer document. **Include your observations in the document. What option would you include to provide a centered header other than the default file name?**

Moving on, lets try fmt on the same file:

12. `fmt smesg`

What happened to the output compared to using pr? Include your answer in the document.

Try:

13. `pr smesg | fmt -s`

What is the observable difference between the last pr command executed and the last one? Include your answer in the document.

The objective of pr and fmt is to reform output into something usable that a printer can accept. Log files like dmesg will output text to the width of the screen and either wrap or truncate the rest. Printers unfortunately do not have that luxury. That is why log output is preprocessed before submitting to a printer.

## *LP Commands*

First off, make sure you have your cups-pdf setup:

14. `apt-get install cups-pdf`
15. `system-config-printer`
16. `lpstat -a`

The PDF printer should be displayed in the printer widget and output from the lpstat command. If not please review the steps 14,15 for a proper installation. If everything is a 'go' execute the following:

17. `lpr -P PDF smesg`

What does the -P option do? Include your answer in the document. Try:

18. `lpr -q -P PDF smesg`
19. `lpq -P PDF`

Record a screenshot, include your answer in the document.

If we take the leap that the first submission was a mistake and want to eliminate it we could:

20. `lprm -P PDF 4`
21. `lpq -P PDF`

Assuming that the job ID was '4'.

You can review what you have accomplished by:

22. `Open a browser window`
23. `http://localhost:631/printers/PDF?which_jobs=all`

This should display all the jobs that have been executed thru the PDF virtual

printer. Note that using PDF you never experienced any output, it was placed in a file specified by \$HOME/PDF. The exact location is dependent on which account you are using when a print command is issued. An `echo $HOME/PDF` will provide the location information currently in effect.

### *Closing remarks*

- Every printer you create is supported by its own queue.
- With the lp commands you can add, remove, redirect print jobs if you have the right rights.
- You can stack jobs for print then print them at your convenience. (Great if you don't want to have your boss reading your resume.)
- Text output can be reformatted to printer limitations as needed. Especially for log files.
- We have only scratched the surface of what is possible. But the above are sufficient for most technicians.

Lab Submissions Proof: Provide screenshots as indicated in the lab; upload your proof to Canvas for grading.

### **Rubric**

#### Checklist/Single Point Mastery

<u>Concerns</u> Working Towards Proficiency	<u>Criteria</u> Standards for This Competency	<u>Accomplished</u> Evidence of Mastering Competency
	Criteria #1: Recorded difference between outputs of two commands (20 points)	
	Criteria #2: Answered question about command that would provide a centered header other than the default file name (20 points)	
	Criteria #3: Answered question What is the observable difference between the last pr command executed and the last one? (20 points)	
	Criteria #4: Answered question about what -P option does in lpg -P Null Printer command (20 points)	
	Criteria #5: Recorded screenshot of lpr -q -P NullPrinter dmesg AND lpq -P NullPrinter commands (20 points)	