UNIX System Administration

Student Workbook

itcourseware
Training Materials For IT Professionals

UNIX System Administration

Jeff Howell and Rob Roselius

Published by ITCourseware, LLC., 7245 South Havana Street, Suite 100, Centennial, CO 80112

Contributing Author: Bruce Batky

Technical Editor: Rob Roselius

Copyeditor: Danielle North

Special thanks to: Many instructors whose ideas and careful review have contributed to the quality of this workbook, including Brandon Caldwell, Rick Feeney, Channing Lovely, and Bob Williams, and the many students who have offered comments, suggestions, criticisms, and insights.

Contents

Chapter 1 - Course Introduction 9

Course Objectives 10
Course Overview 12
Using the Workbook 13
Suggested References 14

Chapter 2 - Overview of System Administration 17

System Administrator's Responsibilities 18
A Brief History of UNIX 20
Evolving Standards 22
Navigating the Documentation 24

Chapter 3 - User Administration 27

What is a "user" in UNIX? 28
The /etc/passwd File 30
Groups 32
The /etc/group File 34
Passwords 36
Adding Users 38
Deleting Users 40
Modifying User Attributes 42
The Login Process 44
/etc/profile and .profile 46
Communicating with Users: /etc/motd 48
Communicating with Users: The wall Command 50
Labs 52

Chapter 4 - File System Basics 55

The Hierarchy 56
Files 58
Directories 60
Device Files 62
Character and Block Devices 64

The /dev Directory 66
Links 68
Symbolic Links 70
A File System Tour 72
The df Command 74
The du Command 76
The find Command 78
Labs 80

Chapter 5 - Advanced File System Concepts 83

The Physical File System 84
The Inode Table 86
File Storage in Disk Blocks 88
The Superblock 90
The Free List 92
Slices and File Systems 94
File System Types 96
Labs 98

Chapter 6 - Disk Management 101

Making a File System 102
The mkfs Command 104
Sharing File Systems 106
The mount Command 108
The fstab File 110
The fsck Command 112
The lost+found Directory 114
The prtvtoc Command 116
Labs 118

Chapter 7 - Backups 121

Backup Strategies 122
Backup Tools 124
The tar Command 126
The cpio Command 128
The dump Command 130
Network Backup Strategies 132
Lab 134

Chapter 8 - UNIX Processes 137

Overview of Processes 138
Process Space 140
Process Table 142
The fork/exec Mechanism 144
The ps Command 146
Background Processes 148
The kill Command 150
Scheduling Jobs 152
The cron Daemon 154
The at Command 156
The crontab Command 158
Format of cron Files 160
Access to Scheduling Facilities 162
Labs 164

Chapter 9 - System Startup and Shutdown 167

Run States (aka run-levels) 168
The init Daemon 170
/etc/inittab 172
The inittab Actions 174
The init Command 176
The rc Scripts 178
Single-User Mode 180
The shutdown Command 182
Labs 184

Chapter 10 - UNIX System Security 187

Security Overview 188
Physical Security 190
Account Security 192
SUID and SGID Settings 194
File and Directory Permissions 196
Software Security 198
Labs 200

Chapter 11 - Performance Monitoring and Tuning 203

Performance Issues 204
Methods of Improving Performance 206
Swapping and Paging 208
The sar Utility 210
Using sar 212
The truss Command 214
Labs 216

Chapter 12 - IP Addressing 219

Basic Network Needs 220
Ethernet (hardware) Addresses 222
IP Addresses 224
DNS vs /etc/hosts to Resolve IP Addresses 226
Network Addresses 228
Network Classes 230
Broadcast Addresses 232
Subnet Masks 234
Labs 236

Chapter 13 - Configuring TCP/IP 239

The /etc/hosts File 240
The ifconfig Command 242
The /etc/services File 244
The inetd Daemon 246
The /etc/inetd.conf File 248
Simple TCP/IP Troubleshooting: The ping Command 250
Simple TCP/IP Troubleshooting: The netstat Command 252
Labs 254

Chapter 14 - The LP Print Service 257

Printing Overview 258
The lp Command 260
The lpstat Command 262
The cancel Command 264
Adding a Printer 266
The lpadmin Command 268
The accept and reject Commands 270

The enable and disable Commands 272
Adding a Networked Printer 274
Other Administrative Commands 276
Labs 278

Appendix A - Network Utilities 281

Network Services 282
telnet - Terminal Emulator 284
ftp - File Transfer 286
ftp Example 288
rcp - Remote Copy 290
rlogin - Remote Login 292
rsh - Remote Commands 294
Labs 296

Appendix B - Kernel Reconfiguration 299

Overview of Reconfiguration 300
Kernel Parameters 302
Steps to Kernel Configuration 304
Specific Steps for SVR4 306
Labs 308

Appendix C - Overview of NIS 311

What is NIS? 312
Why Use NIS? 314
NIS Design and Implementation 316
NIS Maps 318
Configuring NIS 320

Solutions 323

Index 341

CHAPTER 1 - COURSE INTRODUCTION

Course Objectives

- Explain the responsibilities of a UNIX system administrator and perform many of the "hands-on" tasks required to manage a UNIX system.
- Describe the design and operation of UNIX file systems.
- Manage user accounts, including adding and deleting users, changing user account attributes, and controlling password requirements.
- Communicate with users using standard mail and interactive dialogue utilities.
- Manage UNIX processes, including starting and killing jobs in the background, and scheduling jobs to run once or repeatedly at selected times.
- Administer disks and file systems.
- Start up and shut down UNIX systems.
- Administer the print spooling system.
- Give details of UNIX security issues and implement techniques for secure system software and hardware.
- Reconfigure the UNIX kernel to handle new peripherals or add-on software packages.
- Monitor the performance of a multi-user UNIX system and do simple performance tuning operations.

Course Overview

- **Audience**: This is a non-programming course intended for system administration professionals.

- **Prerequisites**: User-level skills in the UNIX environment, such as file manipulation, editing, and the use of utilities are necessary. Also, the ability to read and understand shell scripts is quite helpful.

- **Classroom Environment**:

  - One UNIX workstation per student.

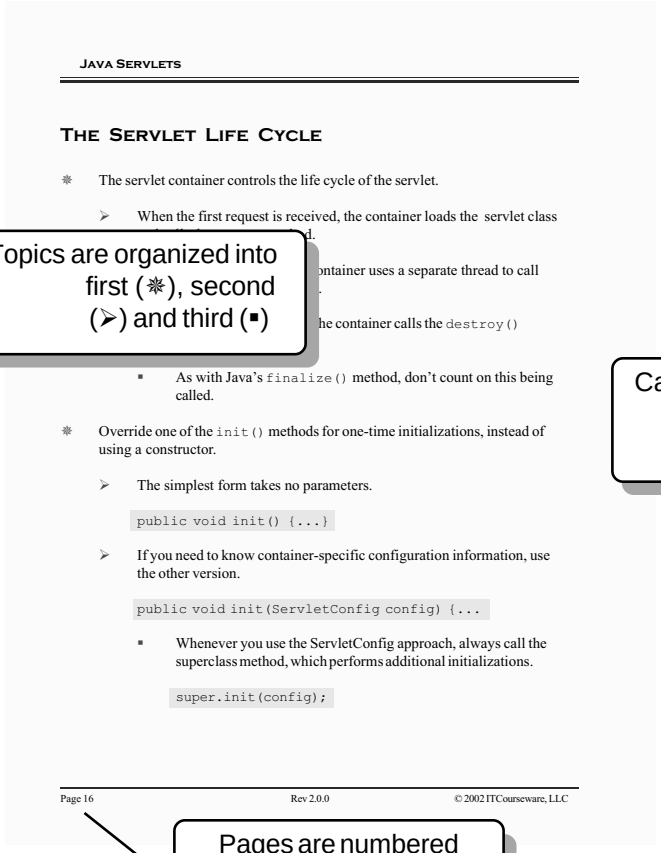
USING THE WORKBOOK

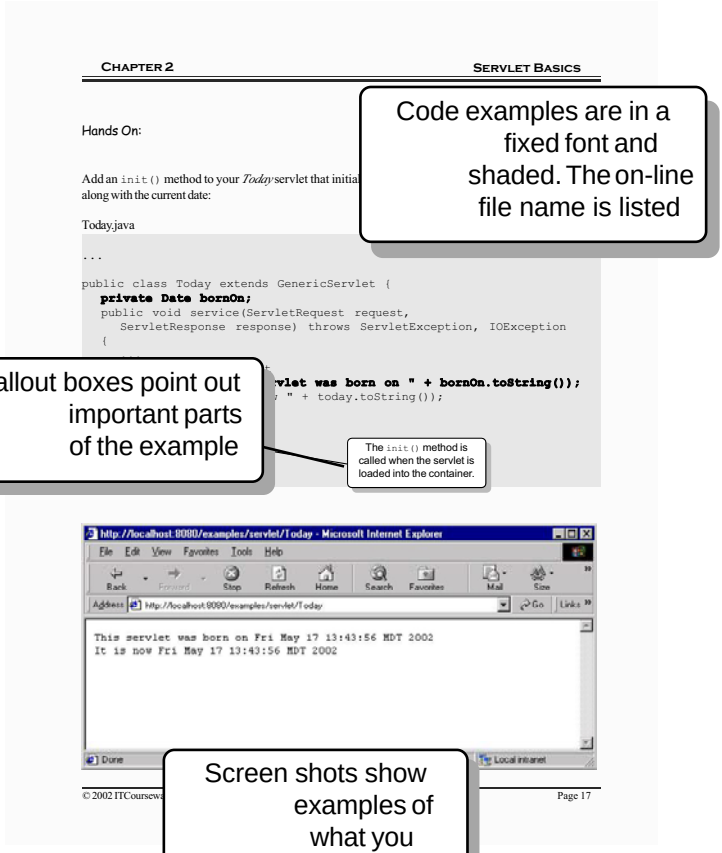
This workbook design is based on a page-pair, consisting of a Topic page and a Support page. When you lay the workbook open flat, the Topic page is on the left and the Support page is on the right. The Topic page contains the points to be discussed in class. The Support page has code examples, diagrams, screen shots and additional information. *Hands On* sections provide opportunities for practical application of key concepts. Try It and Investigate sections help direct individual discovery.

In addition, there is an index for quick look-up. Printed lab solutions are in the back of the book as well as online if you need a little help.

The Topic page provides the main topics for classroom

The Support page has additional information,





Code examples are in a fixed font and shaded. The on-line file name is listed

Topics are organized into first (❋), second (➢) and third (▪)

Callout boxes point out important parts of the example

Screen shots show examples of what you

Pages are numbered sequentially throughout the book, making

Suggested References

Bach, Maurice. 1990. ***The Design of the UNIX Operating System***. Prentice-Hall, Englewood Cliffs, NJ. ISBN 0132017997.

Frisch, AEleen. 2002. ***Essential System Administration, Third Edition***. O'Reilly & Associates, Sebastopol, CA. ISBN 0596003439.

Garfinkle, Simson, Gene Spafford, and Alan Schwartz. 2003. ***Practical UNIX and Internet Security, Third Edition***. O'Reilly & Associates, Sebastopol, CA. ISBN 0596003234.

Hunt, Craig. 2002. ***TCP/IP Network Administration, Third Edition***. O'Reilly & Associates, Sebastopol, CA. ISBN 0596002971.

McKusick, Marshall Kirk, Keith Bostic, Michael J. Karels and John S. Quarterman. 1996. ***The Design and Implementation of the 4.4 BSD Operating System***. Addison-Wesley, Menlo Park, CA. ISBN 0201549794.

Lehtinen, Rick and G.T. Gangemi, Sr. 2006. ***Computer Security Basics, Second Edition***. O'Reilly & Associates, Sebastopol, CA. ISBN 0596006691.

Liu, Cricket and Paul Albitz. 2006. ***DNS and BIND, FifthEdition***. O'Reilly & Associates, Sebastopol, CA. ISBN 0596100574.

Musumeci, Gian-Paolo D. and Michael Loukides. 2002. ***System Performance Tuning, Second Edition***. O'Reilly & Associates, Sebastopol, CA. ISBN 059600284X.

Nemeth, Evi, Garth Snyder and Trent Hein. 2000 ***UNIX System Administration Handbook, Third Edition***. Prentice-Hall, Englewood Cliffs, NJ. ISBN 0130206016.

Robbins, Arnold. 2005. ***UNIX in a Nutshell, Fourth Edition***. O'Reilly & Associates, Sebastopol, CA. ISBN 0596100299.

Eisler, Mike, Ricardo Labiaga, and Hal Stern. 2001. ***Managing NFS and NIS, Second Edition***. O'Reilly & Associates, Sebastopol, CA. ISBN 1565925106.

Wall, Larry, Tom Christiansen and Jon Orwant. 2000. ***Programming Perl, Third Edition.*** O'Reilly & Associates, Sebastopol, CA. ISBN 0596000278.

The most important references for a System Administrator, of course, are the **man** command and your system's documentation set.

CHAPTER 2 - OVERVIEW OF SYSTEM ADMINISTRATION

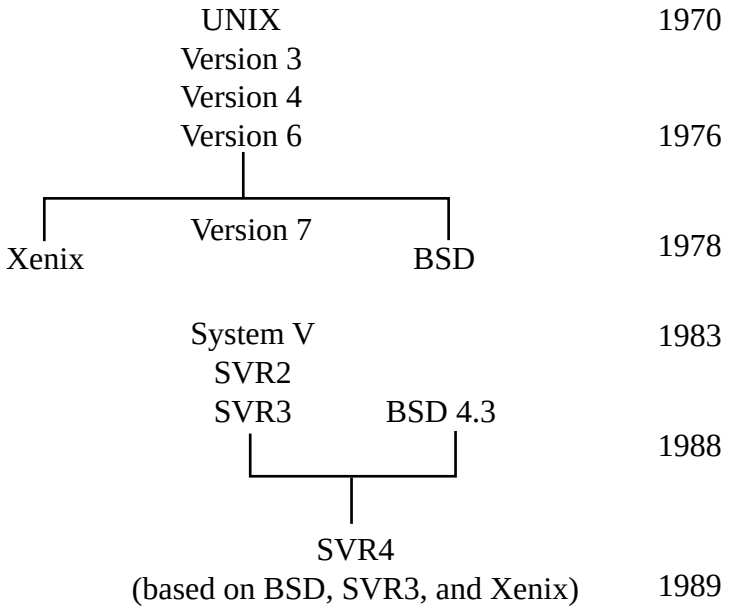
OBJECTIVES

- Understand the responsibilities of a System Administrator.
- Help users understand the role of the System Administrator.
- Understand the history and current status of UNIX.
- Work with the different "flavors" of UNIX.
- Use the UNIX documentation effectively.

System Administrator's Responsibilities

- Ensure the correct operation of the computer system.
  - Set up the system hardware.
  - Install and test the system software.
  - Install and test peripherals.
  - Set up user environments.
  - Install and test application software.
  - Monitor system performance and usage.
  - Manage maintenance contracts.
  - Maintain system logs.
  - Set up and maintain system accounting.
  - Manage application licenses.
- Provide for the safety of the company's assets.
  - Provide for proper physical and logon security.
  - Ensure proper physical environment for hardware.
  - Maintain proper software and data backups.
  - Maintain data archives.
- Responsibilities to the users.
  - Educate and inform users.
  - Respond to user requests.

A BRIEF HISTORY OF UNIX



- Although there have been many derivatives of UNIX, the most recognized versions are based on Xenix, AT&T System V and BSD.

Hardware	Software	OS Based on
Intel	SCO	Xenix
Intel	UnixWare	System V, R4.2
Hewlett-Packard	HP-UX	System V
IBM	AIX	System V, R3.2
Pyramid	DC/OSX	System V
Sequent	Dynix	BSD and System V derivatives
DEC	Ultrix	BSD
SUN	SunOS(4.x)	BSD
SUN	Solaris(2.x)	AT&T SVR4
MIPS	RISC/OS	System V

The MULTICS project, a collaboration between Bell Labs, MIT, and GE, collapsed in 1969. With their free time and expertise, several of the project members, notably Ken Thompson and Dennis Ritchie, began development of a small, simple time-sharing operating system, which they nicknamed UNIX. It ran on a DEC PDP-7.

Between 1973 and 1977, Thompson and Ritchie rewrote the bulk of UNIX in C, with the goal of easy portability. In 1977, they and others succeeded in porting UNIX to several different computer architectures, and in 1979 Bell Labs released UNIX Version 7.

While AT&T used UNIX internally, and licensed it to other companies and to universities, it could not market it as a product. This was due to a 1956 consent decree prohibiting AT&T from participating in the computer market.

In 1979, Thompson taught an Operating Systems class at the University of California at Berkeley (UCB) and left behind a copy of UNIX Version 6. Bill Joy (later the co-founder of Sun Microsystems) and others at UCB began making enhancements to the operating system, and distributed it as BSD (Berkeley Software Distribution) UNIX. Many important enhancements came out of Berkeley.

When AT&T divested its telephone monopoly in 1984, it was allowed to compete in the computer market, and aggressively marketed UNIX. In 1982 it had released a version of UNIX that it called System III; in 1984 it released System V, and subsequent releases have retained this name.

Dozens of UNIX variants proliferated over the years. Xenix, Microsoft's port of UNIX to the microcomputer, was an influential product.

In 1989, AT&T's UNIX System Labs (USL), released System V, Release 4 (SVR4). This release merged features of AT&T, BSD, and Xenix. Sun Microsystems subsequently abandoned its SunOS, a BSD-based UNIX, for SVR4-based Solaris.

In 1993, Novell Corporation bought USL, which had released SVR4.2 — a release geared toward desktop computers. Novell ceded the UNIX trademark to X/Open, and marketed SVR4.2 as UnixWare.

In 1995, SCO took control of SVR4.2 from Novell.

Evolving Standards

- Different types of standards exist:
  - *de jure* Standards
  - *de facto* Standards
  - Open Specifications
  - Public Specifications
- UniForum — An international UNIX users group.
  - Originally called */usr/group*.
  - Produced the 1984 */usr/group* standard.
- SVID — The System V Interface Definition.
  - Heavily influenced by the 1984 */usr/group* standard.
  - Produced by AT&T, starting in the mid-80s.
- POSIX — The Portable Open System Interface standards.
  - Heavily influenced by the 1984 */usr/group* standard and the SVID.
  - Produced and maintained by IEEE working groups.
  - Standards are numbered:
    - 1003.1: System API
    - 1003.2: Shell and Utilities
    - 1003.16: C Binding to 1003.1
    - 1003.17: Directory Services API
- SVR4 — UNIX System V, Release 4
  - POSIX compliant
  - Merged AT&T and BSD UNIX.

Navigating the Documentation

- Section 1 — User Commands
- Section 2 — C System Calls
- Section 3 — C Subroutine Libraries
- Section 4 — File Formats
- Section 5 — Miscellaneous Functions
- Section 7 — Special (Device) Files
- Section 8 — System Administration Commands

CHAPTER 6 - DISK MANAGEMENT

OBJECTIVES

- Describe the steps necessary to create a file system on a disk device.
- Examine and repair a file system.
- Demonstrate proficiency at disk management.

Making a File System

- Format the medium into addressable sectors.
  - Hard drives are typically formatted by the manufacturer and do not need any low-level formatting, or are handled by **install**.
  - Floppy disks are formatted with the **fmtflop** or **format** commands.
- Make a file system on the medium.
  - Specify the file system type.
  - Choose parameters (inodes, logical block size, etc.) appropriate for the type.
- Mount the file system to the directory tree.
  - Create a mount point.
  - Mount the new file system with the **mount** command.
  - Add an entry to */etc/vfstab* (if appropriate).

The mkfs Command

- The **mkfs** command lays out the structure of a file system on a disk slice.
  - Superblock
  - Inode table
  - File system-type-specific structures
- Syntax:

```
# mkfs –F fs_type [options] [–o fs_type-specific options]
                                                raw device.
```

- Since **mkfs** will destroy data residing on the device, it waits 10 seconds before proceeding.
  - **mkfs** can be aborted during this delay with a keyboard interrupt.

Sharing File Systems

- The *Network File System* (NFS) allows you to share file systems among computers.

- The server must **export** or **share** its files system with other machines.

  - A directory tree may be **export**ed or **share**d from the command line.

    ATT:
    BSD:

  - A directory tree may be **export**ed or **share**d at boot.

    ATT:
    BSD:

  - An **export**ed directory tree may be un**export**ed.

    ATT:
    BSD:

- The client must mount the shared file system.

- On System V systems, you may use the **dfshares** command to see what **nfs** file systems are available to mount on a particular client.

The mount Command

- Once it is created, a file system is unavailable to users until it is mounted onto the directory structure.
- A file system is mounted to an existing directory, which is called the mount point of that file system.
- A file system is mounted to the directory tree with the **mount** command.
- Syntax:

```
mount -F fstype -o option1,option2,... block_device mount_point
```

- To remove the mounted file system use the **umount** command.
- Syntax:

```
umount directory
```

or

```
umount block_device_file
```


The fstab File

- To make mounting and unmounting easier, we maintain a datafile describing each file system, with mounting options.
  - */etc/fstab* (SVR3)
  - */etc/vfstab* (SVR4)
  - */etc/checklist* (HP-UX)
- At bootup, the system scans the *fstab* file and attempts to mount everything in it.
- If a file system is described in the *fstab* file, we can mount and unmount it just by naming the mount point.
- The **mountall** command attempts to mount everything in the *fstab* file.

The fsck Command

- Much file I/O is buffered, as is file management.
  - For performance, the system holds in memory copies of the superblock and active inodes.
  - These are only periodically written to disk.
  - If power is cut, the physical file system can be left in an inconsistent state.
- The **fsck** (filesystem check) command examines and attempts to repair the file systems:
- Syntax:

```
fsck –F fstype –y Block_Device
```

  - **fsck** asks the administrator whether it should attempt repairs; the **-y** option tells **fsck** to always assume the answer "yes."
- File systems should be unmounted before being checked with **fsck**. However, the root file system cannot be unmounted.
  - **fsck -m** is automatically run at bootup to determine if each file system was shutdown properly.
  - If **fsck** detects a corrupt file system, a full **fsck** of that file system will be performed.
  - Manual checking of file systems is usually done in single-user mode.

The lost+found Directory

- At times, a directory inode becomes corrupted and loses its pointer to a file or directory inode.
  - At this point that directory has a lost inode.
- During the **fsck** process, these lost inodes are placed in the **lost+found** directory for that file system so they can be retrieved by the system administrator.
- Each file system MUST have a **lost+found** directory.
  - It is usually created by the **mkfs** command when the file system is created.

The prtvtoc Command

- The **prtvtoc** command prints the Volume Table of Contents for a disk device.
- The VTOC describes the disk slices on a device.
  - A slice consists of one or more sectors.
  - Up to 16 slices can be defined on a hard disk, up to 8 on a floppy.
  - Slice 0 refers to the entire device.
- The **edvtoc** command is used to modify the VTOC.
  - First, use **prtvtoc -f** to create a VTOC text file.
  - Modify the text file with an editor (**vi**).
  - Submit the modified file with **edvtoc**.

The **fdisk** command is used to maintain the partition table of a disk.

The partition table defines regions of the disk used by different operating systems. The partition table is in the first sector of the disk.

The maximum number of partitions per disk is 4.

Exactly one partition is the active partition. The computer is booted from the active partition.

fdisk is supported by UNIX, DOS, and OS/2.

LABS

❶ What **mount** options are available for the file system type of your machine's root file system?

❷ Examine the output of the **prtvtoc** command. How many slices on the root disk?

❸ Working together, create a file system on a floppy disk. Mount it to the system's directory tree.

- Format the diskette.
- Use **mkfs** to make the file system.
- Mount the floppy file system to the ***/mnt*** directory.

❹ If you only use **fsck** on unmounted file systems, how do you **fsck** the root file system?

❺ Perform the steps necessary to share your */usr* filesystem with another system in the classroom.

Perform the steps necessary to mount a shared filesystem from another system in the classroom.

CHAPTER 14 - THE LP PRINT SERVICE

OBJECTIVES

- Print using the standard UNIX print spooler.
- Cancel print jobs.
- Add a printer to the UNIX system.

Printing Overview

- Printing on UNIX System V is controlled by the LP Print Service.
- There is functionality for controlling forms, print wheels, and network printers.
- Interface programs exist for filtering print files into a format understood by the printer.
- Unknown **lp** options are passed through, allowing an interface program to use them.

THE lp COMMAND

```
lp [-d printer] filename
```

- Allows the user to send a request to the printer.

```
lp -d laser file.1
```

- **lp** can read standard input:

```
cat /etc/passwd | lp
```

- If successful, **lp** will display a print job number on the screen:

```
request ID is laser-47
```


The lpstat Command

- The **lpstat** command is used to find out information about the LP Print Server and the status of print requests.

- With no options **lpstat** will print the status of all the user's print requests:

```
lpstat
```

- To print the status of all print requests enter:

```
lpstat -a
```


The cancel Command

```
cancel request_ID
```

- Allows the user to cancel a print request.
- Use **lpstat** to get the ***request_ID***.

```
cancel laser-47
```


Adding a Printer

- There are generally 5 steps to add a printer:

1. Physically attach the printer.
   - Varies according to the type of printer (serial, parallel, etc.). Follow manufacturer's instructions.
2. Configure the UNIX device file to which the printer is attached.
   - May require changing ownership and permissions of a device file:

```
# chown lp /dev/tty14
# chgrp lp /dev/tty14
# chmod 660 /dev/tty14
```

   - For serial ports, may need to turn off getty or port monitor.
3. Identify the printer to the LP Print Service with the **lpadmin** command.
4. Tell the print service to accept and queue jobs to the printer with the **accept** command.
5. Tell the print service to send print jobs to the printer with the **enable** command.

On SVR4 systems using the standard printer interface script (default) with an HP laser printer, you must set the **nobanner** parameter to "yes" and put the following line

```
echo "^L"
```

where **^L** is **Ctrl-L**, at the end of the interface script. The script is created in the */usr/spool/lp/admins/lp/interfaces* directory and named the same as the printer you added locally. Put this line immediately before the line that says

```
${DRAIN}
```

THE LPADMIN COMMAND

- The **lpadmin** command is the general administrative tool for managing printers under the LP print service.

- When using **lpadmin** to add or modify printer definitions, always name the specific printer with the **-p** switch.

```
lpadmin -p hplaser2 -v /dev/lp1 -c LASERS -D
"HP Laserjet 4 in main office"
```

- A printer may be accessed through:

  - A direct connection, using a device file, to a serial or parallel port.

  - Over a network, where it is already defined on another system.

  - Over a dialup line, where we supply dialing information as part of the printer information.

- The system default printer is set with the **-d** option**:**

```
# lpadmin -d hplaser2
```

- A printer definition can be removed with the **-x** option:

```
# lpadmin -x oki321
```


The accept and reject Commands

- Only the system adminstrator can tell a printer to **accept** or **reject** requests.
- The **accept** command allows the printer to **accept** print requests from users.

```
accept destinations
```

- The **reject** command prevents requests from being queued to a printer.

```
reject [-r reason] destination
```

- The reason will show up in the **lpstat -a** command.

The enable and disable Commands

- Any user can **enable** or **disable** a printer.
- The **enable** command activates a printer, allowing it to print requests:

```
enable printer
```

- To deactivate a printer, use **disable** to stop printing requests already submitted from **lp**:

```
disable [-r reason] printer
```

  - The reason will show up in the **lpstat -p** command.
  - The printer will still accept queued requests, but it won't print them until the printer is enabled with the **enable** command.

ADDING A NETWORKED PRINTER

- There are two basic steps involved when adding a networked printer to your environment.

  1. On the system without the printer:

     - Use **lpsystem** to specify the system that has the printer.

     - Use **lpadmin** with the **-s** option to make the remote printer available to local users.

  2. On the system with the printer:

     - Use **lpsystem** to allow users on the system without the printer to have access to this system's printer resources.

- Example (the printer is connected to host **bhit9**):

  1. On the system called **bhit10**, type the following to allow **bhit10** users access to a printer called **hplj** on **bhit9**.

     ```
     lpsystem bhit9
     lpadmin -p bhit9lj -s bhit9!hplj
     ```

  2. On **bhit9**, type the following to allow **bhit10** users to use **bhit9** print resources:

     ```
     lpsystem bhit10
     ```


Other Administrative Commands

`lpforms -f {form_name} options`

- Allows the administrator to control forms.

`lpmove`

- To move requests to **dest** printer:

```
lpmove requests dest
```

- To move all requests from **dest1** printer to **dest2** printer:

```
lpmove dest1 dest2
```

- To shut down the LP Print Service:

```
lpshut
```

- To set limits to queue priorities that users can assign to their submitted jobs:

```
lpusers
```

- To set up remote printer information:

```
lpsystem
```


LABS

❶ Define a default printer for your system.

❷ What happens when you send a small text file to the printer, if:

The printer is disabled?
The printer is rejecting requests?

❸ Queue a few jobs to a disabled printer. Cancel them.

❹ Queue a job to a disabled printer. Note the job number.

Go to the print spooler directory. What files are involved in your job?

For exercises 5 and 6, split into two teams working on two servers, each with a printer.

❺ Make your local printer a networked printer available to another team in class.

❻ Provide for spooling from your system to a networked printer on another system.

