

Chapter 8:

Exploring the

UNIX/Linux Utilities

Guide To UNIX Using Linux
Third Edition

Objectives

- Understand many of the UNIX/Linux utilities that are available and how they are classified
- Use the dd utility to copy and convert files
- Make a bootable removable disk
- Monitor hard disk usage

Objectives (continued)

- Use system status utilities
- Monitor and manage processes
- Check the spelling of text in a document
- Use the `cmp` command to compare the contents of two files
- Format text to create and use a man page

Understanding UNIX/Linux Utilities

- UNIX/Linux utilities let you
 - Create and manage files
 - Run programs
 - Produce reports
 - Monitor and maintain the system
 - Recover from a range of errors
- New utilities are continually being added in order to make UNIX/Linux run more efficiently

Understanding UNIX/Linux Utilities (continued)

- Classified into eight major areas:
 - File processing
 - System status
 - Networking
 - Communications
 - Security
 - Programming
 - Source code management
 - Miscellaneous

File Processing Utilities

Table 8-1 File-processing utilities

Command	Brief Description of Function
<i>awk</i>	Processes files
<i>cat</i>	Displays files (and is used with other tools to concatenate files)
<i>cmp</i>	Compares two files
<i>comm</i>	Compares sorted files, and shows differences
<i>cp</i>	Copies files
<i>cpio</i>	Copies and backs up files to an archive
<i>cut</i>	Selects characters or fields from input lines
<i>dd</i>	Copies and converts input records
<i>diff</i>	Compares two text files, and shows differences
<i>dump</i>	Backs up files
<i>fdformat</i>	Formats a floppy disk at a low level
<i>file</i>	Displays the file type
<i>find</i>	Finds files within file tree
<i>fmt</i>	Formats text very simply
<i>grep</i>	Matches patterns in a file
<i>groff</i>	Processes embedded text formatting codes
<i>gzip</i>	Compresses or decompresses files
<i>head</i>	Displays the first part of a file (first 10 lines by default)
<i>ispell</i>	Checks one or more files for spelling errors
<i>less</i>	Displays files allowing for scrolling forward and backward (pauses when screen is full)
<i>ln</i>	Creates a link to a file
<i>lpr</i>	Sends a file to a printer or printer device
<i>ls</i>	Lists file and directory names and attributes
<i>man</i>	Displays documentation for commands
<i>mkbootdisk</i>	Creates a floppy boot disk from which to boot a system
<i>mkdir</i>	Creates a new directory
<i>mkfs</i>	Builds a UNIX/Linux file system
<i>mount</i>	Mounts file systems and devices
<i>mv</i>	Renames and moves files and directories
<i>newfs</i>	Creates a new file system (used in UNIX systems in particular)
<i>od</i>	Formats and displays data from a file in octal, hexadecimal, and ASCII formats
<i>paste</i>	Concatenates files horizontally
<i>pr</i>	Formats text files for printing and displays them
<i>pwd</i>	Shows the directory you are in
<i>rdev</i>	Queries or sets the root image device
<i>restore</i>	Restores files (from a dump)
<i>rm</i>	Removes files
<i>rmdir</i>	Removes directories
<i>sed</i>	Edits streams (noninteractive)
<i>sort</i>	Sorts or merges files
<i>tail</i>	Displays the last lines of files (last 10 lines by default)

File Processing Utilities (continued)

Table 8-1 File-processing utilities (continued)

Command	Brief Description of Function
<i>tar</i>	Copies and backs up files to a tape archive
<i>touch</i>	Changes file modification dates
<i>uniq</i>	Displays unique lines of sorted file
<i>wc</i>	Counts lines, words, and bytes
<i>whereis</i>	Locates information about a specific file

The tar command

- Tar stands for tape archiver. It is used as a powerful backup and restore utility.
- Most Linux files are downloaded as .tar files.
- `tar -xvf file1`
 - x extracts files
 - v verbose
 - f filename

System Status Utilities

Table 8-2 System status utilities

Command	Brief Description of Function
<i>date</i>	Sets and displays date and time
<i>df</i>	Displays the amount of free space remaining on disk
<i>du</i>	Summarizes file space usage
<i>file</i>	Determines file type (for example: shell script, executable, ASCII text, and others)
<i>finger</i>	Displays detailed information about users who are logged in
<i>free</i>	Displays amount of free and used memory in the system
<i>edquota</i>	Displays user disk quotas and enables them to be changed
<i>kill</i>	Terminates a running process
<i>ps</i>	Displays process status by process identification number and name
<i>sleep</i>	Suspends process execution for a specified time
<i>top</i>	Dynamically displays the status of processes in real time, focusing on those processes that are using the most CPU resources
<i>uname</i>	Shows information about the operating system
<i>vmstat</i>	Shows information about virtual memory use
<i>w</i>	Displays detailed information about the users who are logged in
<i>who</i>	Displays brief information about the users who are logged in

The finger and sleep command

- The finger command can be used to find out information about users.
- finger username
 - Displays information about the user including username, full name, home directory, last login time, shell, etc.
- sleep
 - Suspends the execution of the process for a specified time.

Network Utilities

Table 8-3 Network utilities

Command	Brief Description of Function
<i>ftp</i>	Transfers files over a network
<i>ifconfig</i>	Sets up a network interface
<i>netstat</i>	Shows network connection information

Table 8-3 Network utilities (continued)

Command	Brief Description of Function
<i>nfsstat</i>	Shows statistics for Network File System (NFS; file upload and download) activity
<i>ping</i>	Polls another network station (using TCP/IP); great for a fast determination about whether your network connection is working
<i>rcp</i>	Remotely copies a file from a network computer
<i>rlogin</i>	Logs in to a remote computer
<i>route</i>	Displays routing table information, and can be used to configure routing
<i>rsh</i>	Executes commands on a remote computer
<i>showmount</i>	Lists clients that have mounted volumes on a server
<i>telnet</i>	Connects to a remote computer on a network
<i>wvdial</i>	Controls a modem dialer for dial-up connections over a phone line

Network Utility commands

- `ifconfig` – sets up a network interface card
 - Can be used to troubleshooting networking
- `netstat` – shows network connection information
- `ping` – establishes connectivity to a remote device
- `route` – displays routing table information

Communications Utilities

Table 8-4 Communications utilities

Command	Brief Description of Function
<i>mail</i>	Sends electronic mail messages
<i>mesg</i>	Denies (<i>mesg n</i>) or accepts (<i>mesg y</i>) messages
<i>talk</i>	Lets users simultaneously type messages to each other
<i>wall</i>	Sends a message to all logged in users (who have permissions set to receive messages)
<i>write</i>	Sends a message to another user

Communications Utilities

- mail – sends e-mail
- mesg n – denies any real-time messages
- mesg y – accepts any real-time messages
- talk – allows users to simultaneously ‘chat’ with other logged in users.
- wall – sends a message to all logged-in users.
- write – sends a message to a user

Security Utilities

Table 8-5 Security utilities

Command	Brief Description of Function
<i>chgrp</i>	Changes the group associated with a file or the file's group ownership
<i>chmod</i>	Changes the access permissions of a file or directory
<i>chown</i>	Changes the owner of a file

Table 8-5 Security utilities (continued)

Command	Brief Description of Function
<i>ipchains</i>	Manages a firewall and packet filtering (do not use if you are using <i>iptables</i> instead)
<i>iptables</i>	Manages a firewall and packet filtering (do not use if you are using <i>ipchains</i> instead)
<i>passwd</i>	Changes a password

Programming and Source Code Management Utilities

Table 8-6 Programming utilities

Command	Brief Description of Function
<i>configure</i>	Configures program source code automatically
<i>g++</i>	Compiles a C++ program
<i>gcc</i>	Compiles a C program
<i>make</i>	Maintains program source code
<i>patch</i>	Updates source code

Table 8-7 Source code management utilities

Command	Brief Description of Function
<i>ci</i>	Creates changes in Revision Control Systems (RCS)
<i>co</i>	Retrieves an unencoded revision of an RCS file
<i>cvs</i>	Manages concurrent access to files in a hierarchy
<i>rcs</i>	Creates or changes the attributes of an RCS file
<i>rlog</i>	Prints a summary of the history of an RCS file

Security Utilities

- `chgrp` – changes the default group associated with a file
- `chmod` – changes the access permissions of a file or directory
- `chown` – changes the owner of a file or directory

Miscellaneous Utilities

Table 8-8 Miscellaneous utilities

Command	Brief Description of Function
<i>at</i>	Executes a command or script at a specified time
<i>atq</i>	Shows the jobs (commands or scripts) already scheduled to run
<i>atrm</i>	Enables you to remove a job (command or script) that is scheduled to run
<i>batch</i>	Runs a command or script, and is really a subset of the <i>at</i> command that takes you to the <i>at></i> prompt, if you type only <i>batch</i> (in Fedora and Red Hat Enterprise Linux, a command or script is run when the system load is at an acceptable level)
<i>cal</i>	Displays a calendar for a month or year
<i>cd</i>	Changes to a directory
<i>crontab</i>	Schedules a command to run at a preset time
<i>expr</i>	Evaluates expressions (used for arithmetic and string manipulations)
<i>fsck</i>	Checks and fixes problems on a file system (repairs damage)
<i>printenv</i>	Prints environment variables
<i>tee</i>	Clones output stream to one or more files
<i>tr</i>	Replaces specified characters (a translation filter)
<i>tty</i>	Displays terminal path name
<i>xargs</i>	Converts standard output of one command into arguments for another

Using the dd Command

- Allows you to copy a file and change the format of the destination file
- Has a rich set of options to handle copies when other methods are inappropriate such as when the format of the destination file needs to be altered. (ASCII to EBCDIC, uppercase to lowercase, etc.)
- An advantage to using the dd command over cp is that all users, not just the administrator, can copy files to and from the floppy drive without mounting it.

The dd command

- Options

if= input file

of=output file

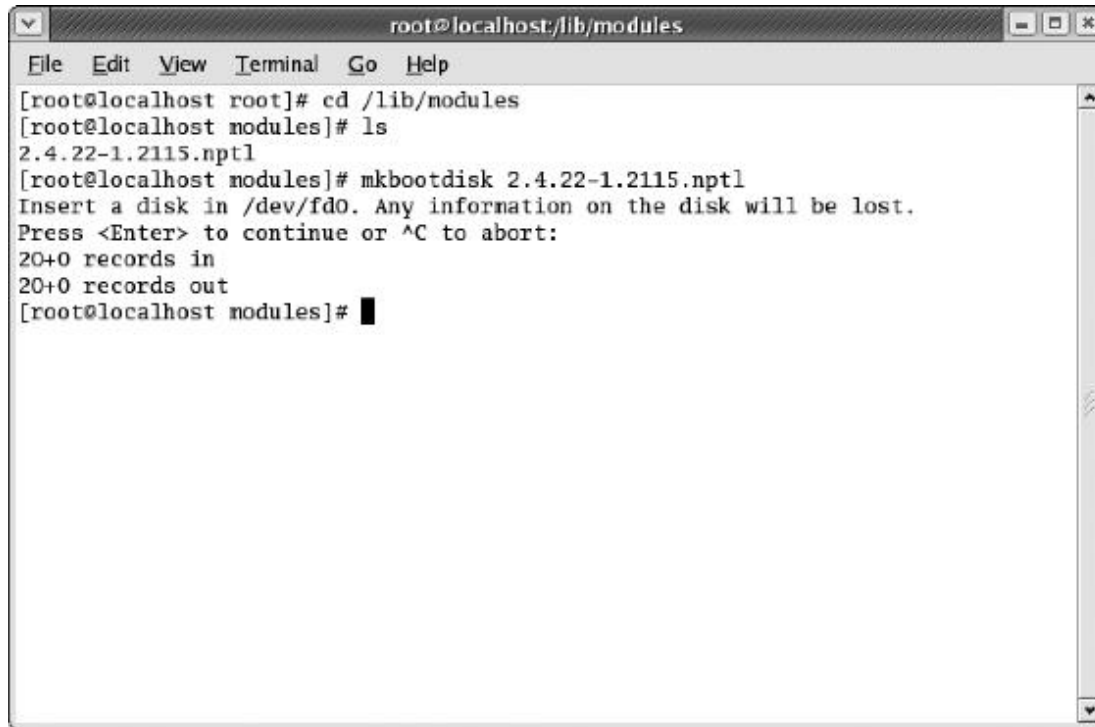
conv=ascii converts destination to ascii

conv=lc case converts uppercase to lower

Making a Bootable Removable Disk

- Make a bootable floppy disk because a computer problem may prevent you from starting UNIX/Linux from the system
- Bootable floppy disks or CD-ROMs can be made using utilities provided by your version of UNIX/Linux
 - Often the mkbootdisk command

Making a Bootable Removable Disk (continued)

A terminal window titled 'root@localhost:/lib/modules' with a menu bar (File, Edit, View, Terminal, Go, Help). The terminal shows the following commands and output:

```
[root@localhost root]# cd /lib/modules
[root@localhost modules]# ls
2.4.22-1.2115.nptl
[root@localhost modules]# mkbootdisk 2.4.22-1.2115.nptl
Insert a disk in /dev/fd0. Any information on the disk will be lost.
Press <Enter> to continue or ^C to abort:
20+0 records in
20+0 records out
[root@localhost modules]#
```

Find the kernel
version and
create a floppy
boot disk

Figure 8-1 Creating a floppy boot disk

Checking Hard Disk Usage

- To maintain adequate hard disk free space, use these strategies:
 - Be vigilant against running dangerously low on free space by using the `df` command
 - Watch for conspicuous consumption using the `du` command
 - Follow a routine schedule for “garbage” collection and removal by using the `find` and `rm` commands

Using the df (disk free) Utility

The df utility reports on the status of 1024-byte blocks that are allocated, used, and available and the mount point

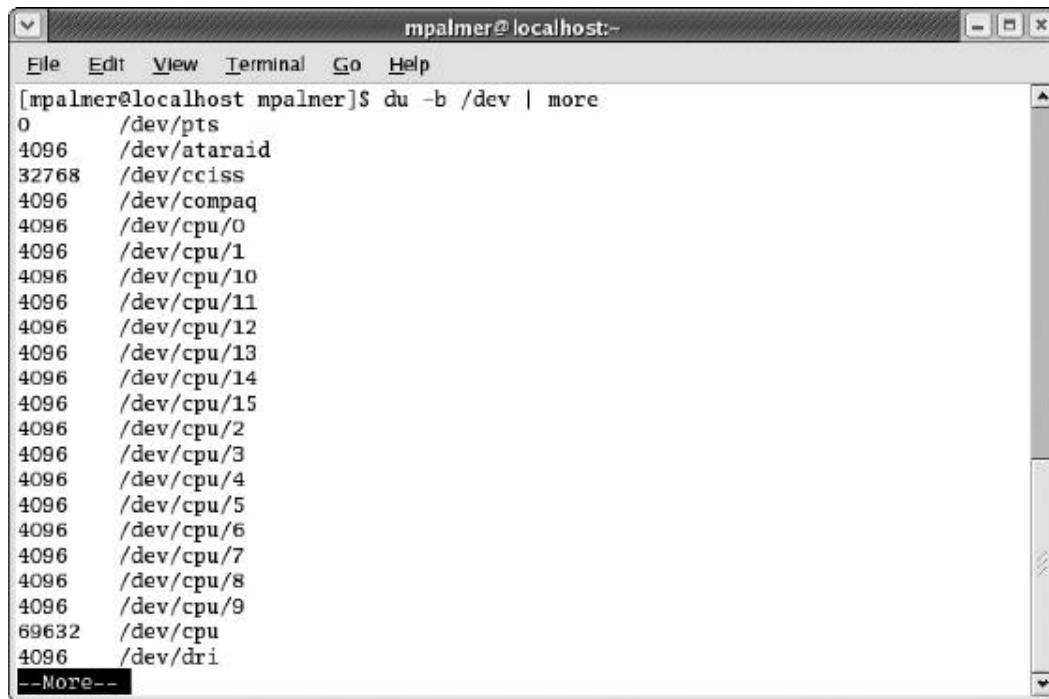
- h human readable form
- m sizes in megabytes



```
mpalmer@localhost:~  
File Edit View Terminal Go Help  
[mpalmer@localhost mpalmer]$ df /dev/hda5 -m  
Filesystem      1M-blocks      Used Available Use% Mounted on  
/dev/hda5        33139        2491      28965    8% /  
[mpalmer@localhost mpalmer]$
```

Figure 8-2 Viewing information for one file system in megabytes

Using the du (disk usage) Utility



```
mpalmer@localhost:~  
File Edit View Terminal Go Help  
[mpalmer@localhost mpalmer]$ du -b /dev | more  
0      /dev/pts  
4096   /dev/ataraid  
32768  /dev/cciss  
4096   /dev/compaq  
4096   /dev/cpu/0  
4096   /dev/cpu/1  
4096   /dev/cpu/10  
4096   /dev/cpu/11  
4096   /dev/cpu/12  
4096   /dev/cpu/13  
4096   /dev/cpu/14  
4096   /dev/cpu/15  
4096   /dev/cpu/2  
4096   /dev/cpu/3  
4096   /dev/cpu/4  
4096   /dev/cpu/5  
4096   /dev/cpu/6  
4096   /dev/cpu/7  
4096   /dev/cpu/8  
4096   /dev/cpu/9  
69632  /dev/cpu  
4096   /dev/dri  
--More--
```

Figure 8-3 Viewing *du* information for the */dev* directory

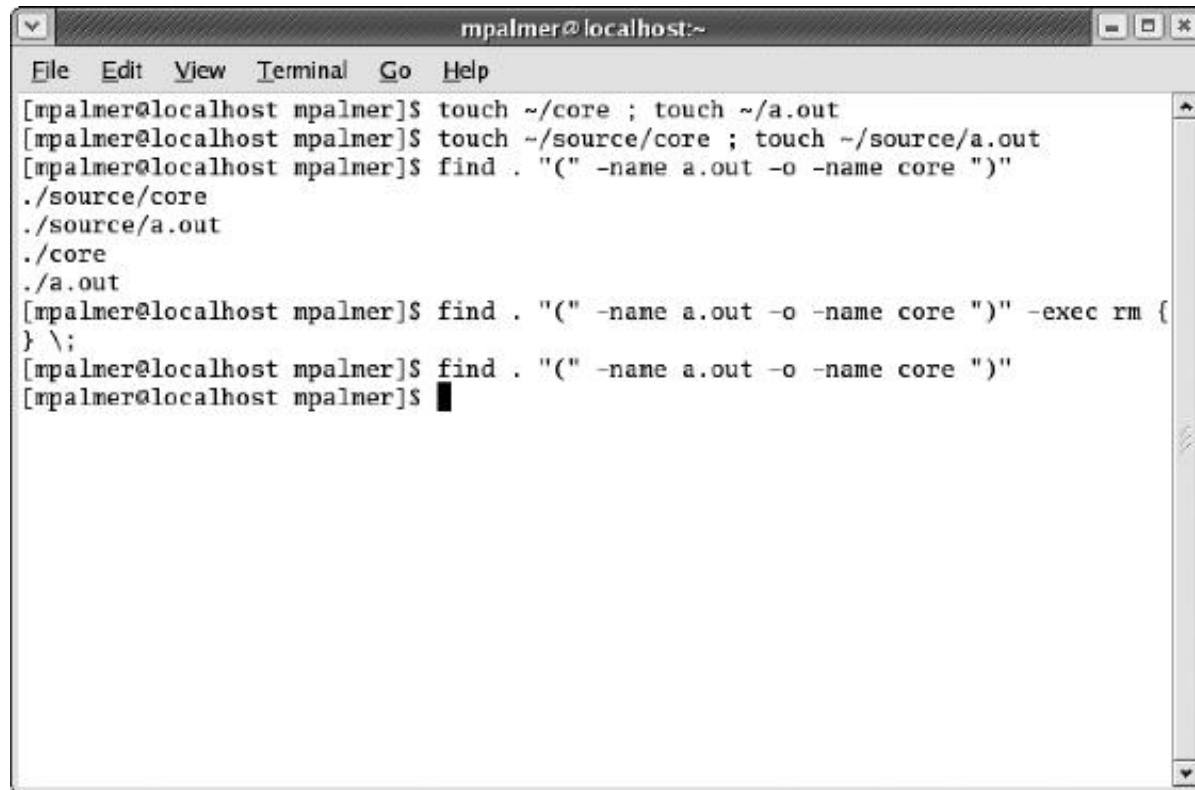
The *du* utility summarizes disk usage, expressed in 512-byte blocks (default) or by the number of bytes (-b option)

- a displays info for files/dirs
- c creates an ending total
- b displays in bytes

Removing Garbage Files

- Garbage files are temporary files that lose their usefulness after several days
- Two examples of garbage files are core files (named core) and a.out files
- Use the find command to assist you in locating these files and the rm command to remove them.
 - In the following slide, find is used to remove garbage files. The `–exec rm {} \;` option tells Linux to rm all files found `{}` by the command.

Removing Garbage Files (continued)

A terminal window titled 'mpalmer@localhost:~' with a menu bar (File, Edit, View, Terminal, Go, Help). The terminal shows a sequence of commands to create and then delete files. First, 'touch ~/core ; touch ~/a.out' is executed. Then, 'touch ~/source/core ; touch ~/source/a.out' is executed. Next, 'find . "(" -name a.out -o -name core ")"' is run, listing './source/core', './source/a.out', './core', and './a.out'. Finally, 'find . "(" -name a.out -o -name core ")" -exec rm { } \;' is executed to delete these files. The prompt returns, and the user presses the Enter key.

```
mpalmer@localhost:~  
File Edit View Terminal Go Help  
[mpalmer@localhost mpalmer]$ touch ~/core ; touch ~/a.out  
[mpalmer@localhost mpalmer]$ touch ~/source/core ; touch ~/source/a.out  
[mpalmer@localhost mpalmer]$ find . "(" -name a.out -o -name core ")"  
./source/core  
./source/a.out  
./core  
./a.out  
[mpalmer@localhost mpalmer]$ find . "(" -name a.out -o -name core ")" -exec rm {  
} \;  
[mpalmer@localhost mpalmer]$ find . "(" -name a.out -o -name core ")"  
[mpalmer@localhost mpalmer]$
```

Figure 8-10 Using the *find* command to delete garbage files

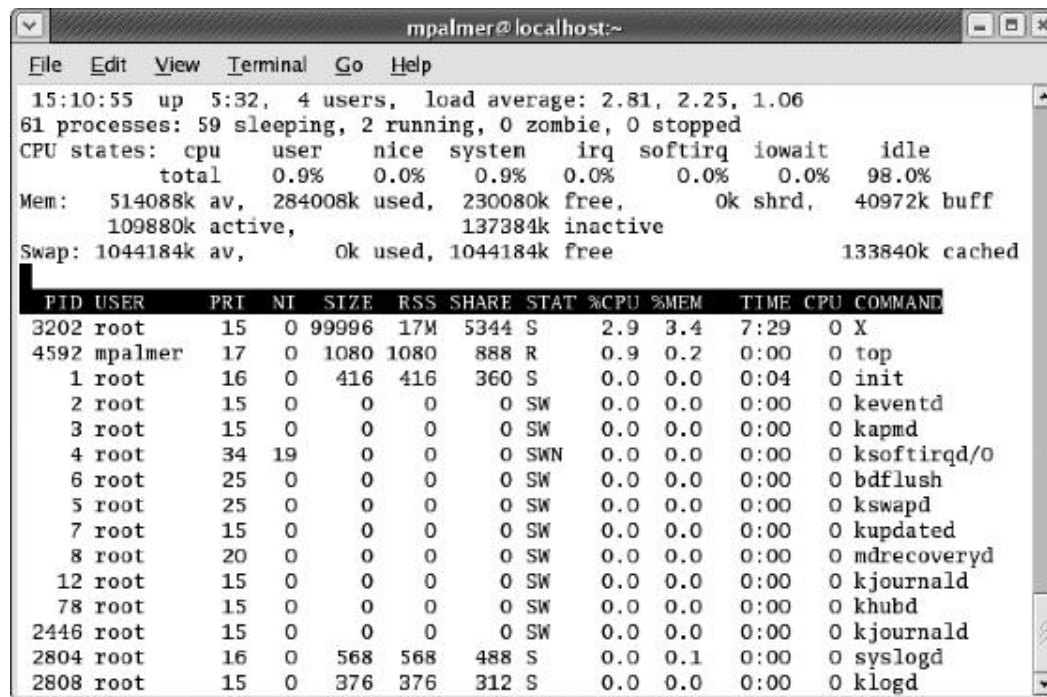
Using System Status Utilities

- System status commands reflect the system's performance
- System engineers primarily use the data related to system status
- Good to know how to obtain and store relevant information to send to system administrator and tune-up specialists

Using the top Command

- One of the most effective utilities for auditing system performance is the top command
- The top command displays a listing of the most CPU-intensive tasks in real time
- Updates every five seconds by default

Using the top Command (continued)



The screenshot shows a terminal window titled 'mpalmer@localhost:~'. The output of the 'top' command is displayed. At the top, it shows system statistics: '15:10:55 up 5:32, 4 users, load average: 2.81, 2.25, 1.06'. Below this, it shows process statistics: '61 processes: 59 sleeping, 2 running, 0 zombie, 0 stopped'. The CPU states are listed as: 'CPU states: cpu user nice system irq softirq iowait idle', with a total of '0.9% 0.0% 0.9% 0.0% 0.0% 0.0% 98.0%'. Memory usage is shown as: 'Mem: 514088k av, 284008k used, 230080k free, 0k shrd, 40972k buff', with '109880k active, 137384k inactive'. Swap usage is shown as: 'Swap: 1044184k av, 0k used, 1044184k free, 133840k cached'. Below the statistics is a table of processes.

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	CPU	COMMAND
3202	root	15	0	99996	17M	5344	S	2.9	3.4	7:29	0	X
4592	mpalmer	17	0	1080	1080	888	R	0.9	0.2	0:00	0	top
1	root	16	0	416	416	360	S	0.0	0.0	0:04	0	init
2	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	keventd
3	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	kapmd
4	root	34	19	0	0	0	SWN	0.0	0.0	0:00	0	ksoftirqd/0
6	root	25	0	0	0	0	SW	0.0	0.0	0:00	0	bdfldush
5	root	25	0	0	0	0	SW	0.0	0.0	0:00	0	kswapd
7	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	kupdated
8	root	20	0	0	0	0	SW	0.0	0.0	0:00	0	mdrecoveryd
12	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	kjournald
78	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	khubd
2446	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	kjournald
2804	root	16	0	568	568	488	S	0.0	0.1	0:00	0	syslogd
2808	root	15	0	376	376	312	S	0.0	0.0	0:00	0	klogd

The top utility run without any options specified

Figure 8-11 Sample *top* display

Using the uptime Command

- Uptime tells you how long a system has been running since the last time it was booted
- Displays current time, how long the system has been up, number of users on the system, and the load average for 1, 5, and 15 minutes

Using the free Command

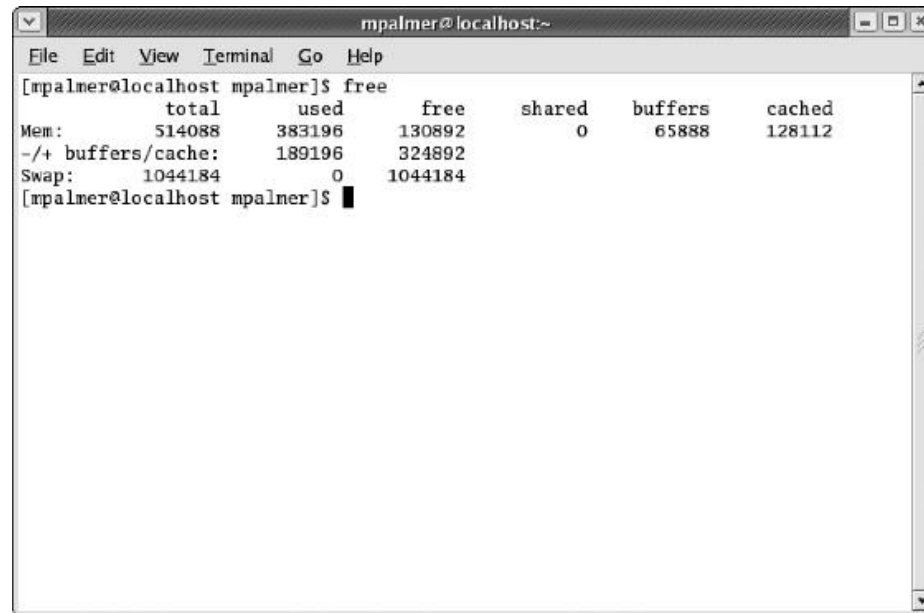
The free utility displays the amount of free and used memory in the system

-b bytes

-m megabytes

-g gigabytes

-t totals

A terminal window titled 'mpalmer@localhost:~' with a menu bar (File, Edit, View, Terminal, Go, Help). The command 'free' has been executed, showing memory usage statistics. The output is as follows:

```
[mpalmer@localhost mpalmer]$ free
              total        used        free      shared    buffers     cached
Mem:          514088      383196      130892           0         65888      128112
-/+ buffers/cache:    189196      324892
Swap:        1044184           0      1044184
```

Figure 8-4 Using the *free* command to monitor memory and swap usage

Forwarding top and free Output

- When problems arise with performance, may need to forward top and free output to support person
- Use redirection (>) to store outputs in files
 - `top n 3 > topdata`

Managing Processes

- A process is identified through a unique number called a process id (pid)
- Unix/Linux offer utilities to run, monitor, and kill processes using pids

Running Processes in the Background

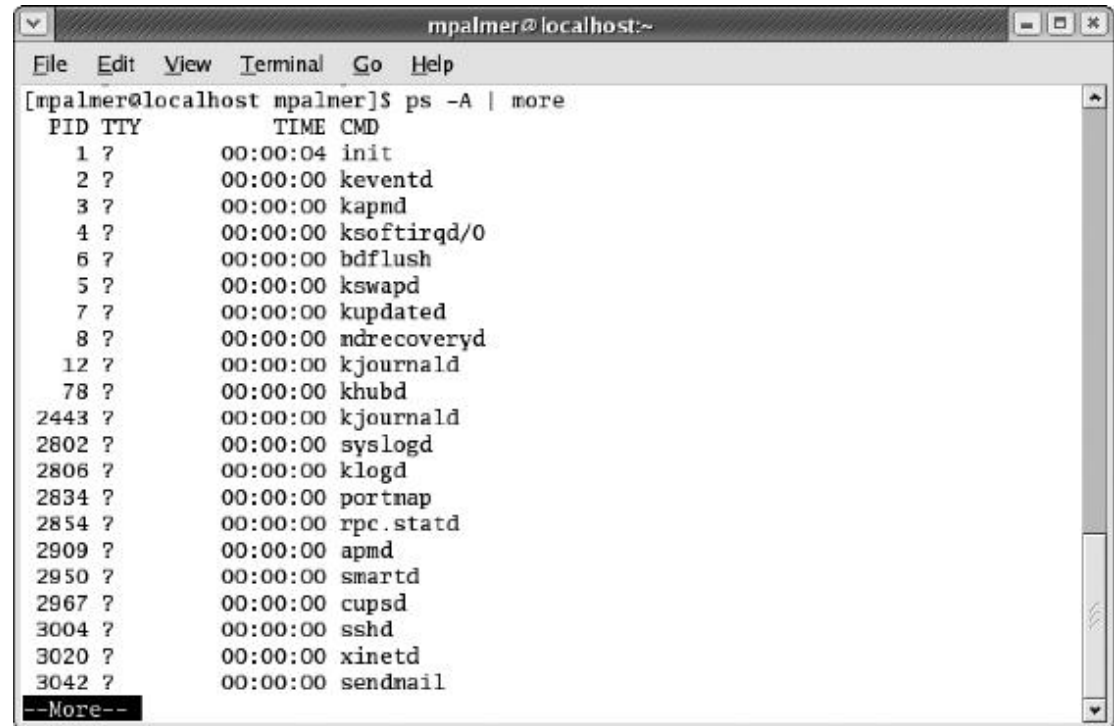
- Can run a process in the background while working with another program in the foreground
- To run a program in the background, append the & character to end of the startup command, e.g., `top&`

Monitoring Processes

The `ps` command with the `-A` option shows a list of all system processes currently running

`ps -aux`

is used to display all of the processes running on the system

A terminal window titled 'mpalmer@localhost:~' showing the output of the 'ps -A' command. The output is a table with columns: PID, TTY, TIME, and CMD. It lists various system processes like init, keventd, ksysd, ksoftirqd/0, bdfush, kswapd, kupdated, ndrecoveryd, kjournald, khubd, ksysd, syslogd, klogd, portnap, rpc.statd, apmd, smartd, cupsd, sshd, xinetd, and sendmail. The list is truncated with '--More--' at the bottom.

PID	TTY	TIME	CMD
1	?	00:00:04	init
2	?	00:00:00	keventd
3	?	00:00:00	ksysd
4	?	00:00:00	ksoftirqd/0
6	?	00:00:00	bdfush
5	?	00:00:00	kswapd
7	?	00:00:00	kupdated
8	?	00:00:00	ndrecoveryd
12	?	00:00:00	kjournald
78	?	00:00:00	khubd
2443	?	00:00:00	kjournald
2802	?	00:00:00	syslogd
2806	?	00:00:00	klogd
2834	?	00:00:00	portnap
2854	?	00:00:00	rpc.statd
2909	?	00:00:00	apmd
2950	?	00:00:00	smartd
2967	?	00:00:00	cupsd
3004	?	00:00:00	sshd
3020	?	00:00:00	xinetd
3042	?	00:00:00	sendmail

Figure 8-5 Viewing all of the running processes

Killing Processes

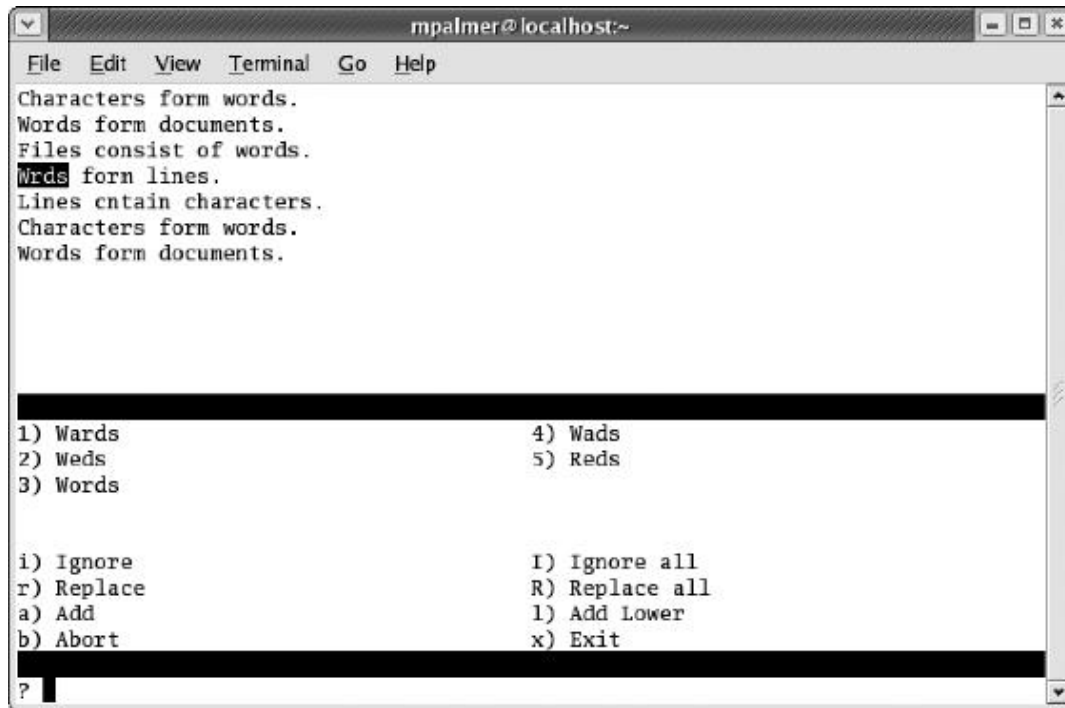
- Administrator with root privileges can kill any user's processes
- User can kill owned processes
- Use kill command with the pid of the process
- Use kill -9 (sure kill) to stop a process that doesn't respond to an initial kill command
- If I have started executing a program (p1) that is running infinitely, I may kill that process with the following steps:

ps

(Note the pid number of the process p1- we'll assume it is 608)

kill 608

Checking the Spelling of a Document



ispell scans a document, displays errors on the screen and suggests alternative spellings

Figure 8-6 Checking the spelling in a document with *ispell*

Comparing Files

- Use the `cmp` utility to compare the contents of two files, and report the first difference between them
- The `cmp` command displays the position and line number of this difference
- If there are no differences, the `cmp` command displays nothing

Formatting Text in UNIX/Linux

- Text formatting in UNIX/Linux involves preparing a text file with embedded typesetting commands and then processing the file
- UNIX's nroff and troff commands were the early standard in formatting programs
- An embedded code is a special sequence of characters that is included with the regular text of the file

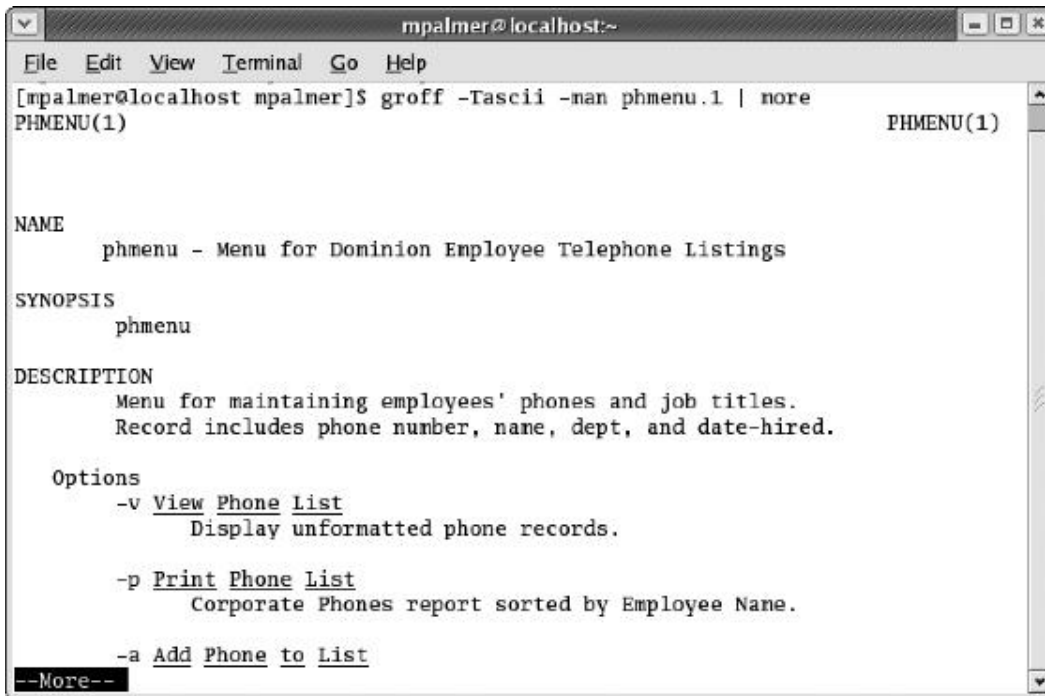
Formatting Text in UNIX/Linux (continued)

Table 8-9 Sample *groff* embedded commands

Embedded Command	Meaning
.ce <i>n</i>	Center next <i>n</i> lines
.ds C	Center
.ds R	Right-justify
.p <i>n</i>	Start a new paragraph indented <i>n</i> characters
.sa 0	Turn off justification
.sa 1	Turn on justification
.ul <i>n</i>	Underline the next <i>n</i> lines

Linux introduced *groff*, which implements the features of both *nroff* and *troff*

Formatting Text in UNIX/Linux (continued)

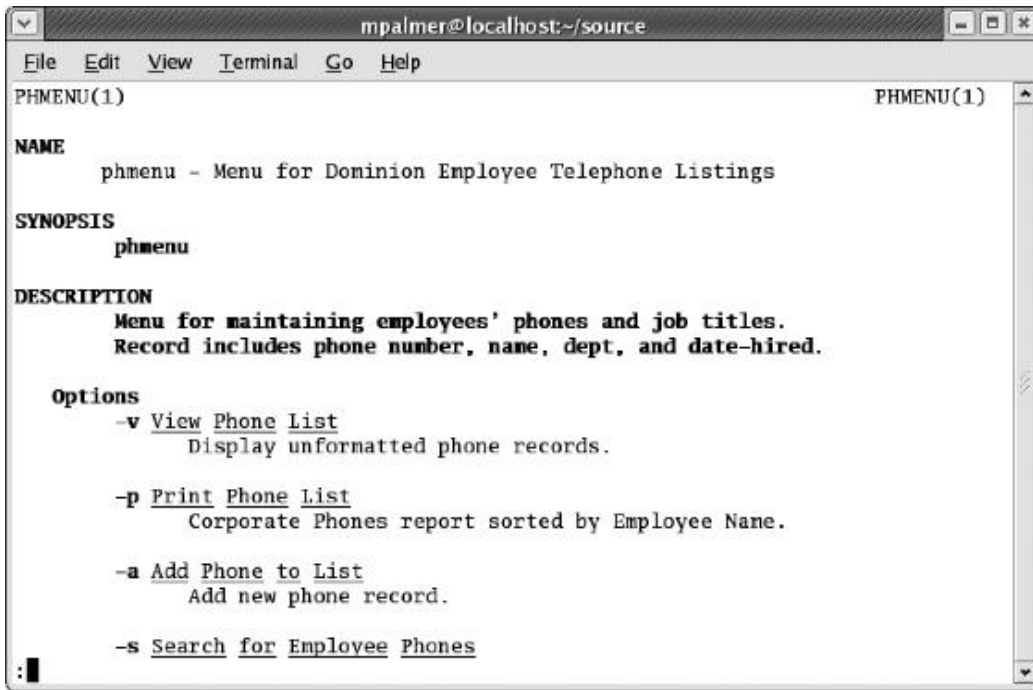


```
mpalmer@localhost:~  
File Edit View Terminal Go Help  
[mpalmer@localhost mpalmer]$ groff -Tascii -nan phmenu.1 | more  
PHMENU(1) PHMENU(1)  
  
NAME  
    phmenu - Menu for Dominion Employee Telephone Listings  
  
SYNOPSIS  
    phmenu  
  
DESCRIPTION  
    Menu for maintaining employees' phones and job titles.  
    Record includes phone number, name, dept, and date-hired.  
  
Options  
    -v View Phone List  
        Display unformatted phone records.  
  
    -p Print Phone List  
        Corporate Phones report sorted by Employee Name.  
  
    -a Add Phone to List  
--More--
```

Groff can be used to produce a man page that contains the standard man page sections

Figure 8-17 Using *groff* to view the phmenu documentation

Formatting Text in UNIX/Linux (continued)



```
mpalmer@localhost:~/source
File Edit View Terminal Go Help
PHMENU(1) PHMENU(1)
NAME
    phmenu - Menu for Dominion Employee Telephone Listings
SYNOPSIS
    phmenu
DESCRIPTION
    Menu for maintaining employees' phones and job titles.
    Record includes phone number, name, dept, and date-hired.
Options
    -v View Phone List
        Display unformatted phone records.
    -p Print Phone List
        Corporate Phones report sorted by Employee Name.
    -a Add Phone to List
        Add new phone record.
    -s Search for Employee Phones
```

Man pages are made available to others by having a privileged user copy it to one of the man page directories

Figure 8-18 Using *man* to display the phmenu documentation

Chapter Summary

- UNIX/Linux utilities are classified into eight major functional areas
- Utility programs are called commands: executed by entering names on the command line
- dd command options allow it to handle copies when other copying methods fail
- To make a bootable removable disk, use provided utilities such as mkbootdisk

Chapter Summary (continued)

- `df` checks and reports on free disk space
- `du` checks for disk usage
- Use `find` to retrieve temporary files and use `rm` to remove them
- `top` and `free` provide detailed views of the “internals” of the system that can be redirected to a file for system tune-up

Chapter Summary (continued)

- Run a program in the background by appending & to the end of a command
- ps displays all running processes
- kill terminates a specific process
- ispell scans for spelling errors
- Text formatting involves
 - Embedding typesetting commands in a file
 - Processing the file with a program that generates commands for the output device

Chapter Summary (continued)

- Linux introduced groff, which implements the features of both nroff and troff
- Text formatted with groff can be used to create new man pages