A Guide to Unix Using Linux Fourth Edition

Chapter 1

The Essence of UNIX and Linux

Objectives

- Explain operating systems, including PC and server operating systems
- Describe the UNIX and Linux operating systems
- Explain the purpose of UNIX/Linux shells
- Understand how to select user names and passwords
- Connect to UNIX/Linux using Telnet or SSH

Objectives (continued)

- Use basic UNIX/Linux commands and command-line editing features
- Explain the role of a system administrator
- Change your password for security
- Use multiple commands to view the contents of files
- Redirect output to a file

Understanding Operating Systems

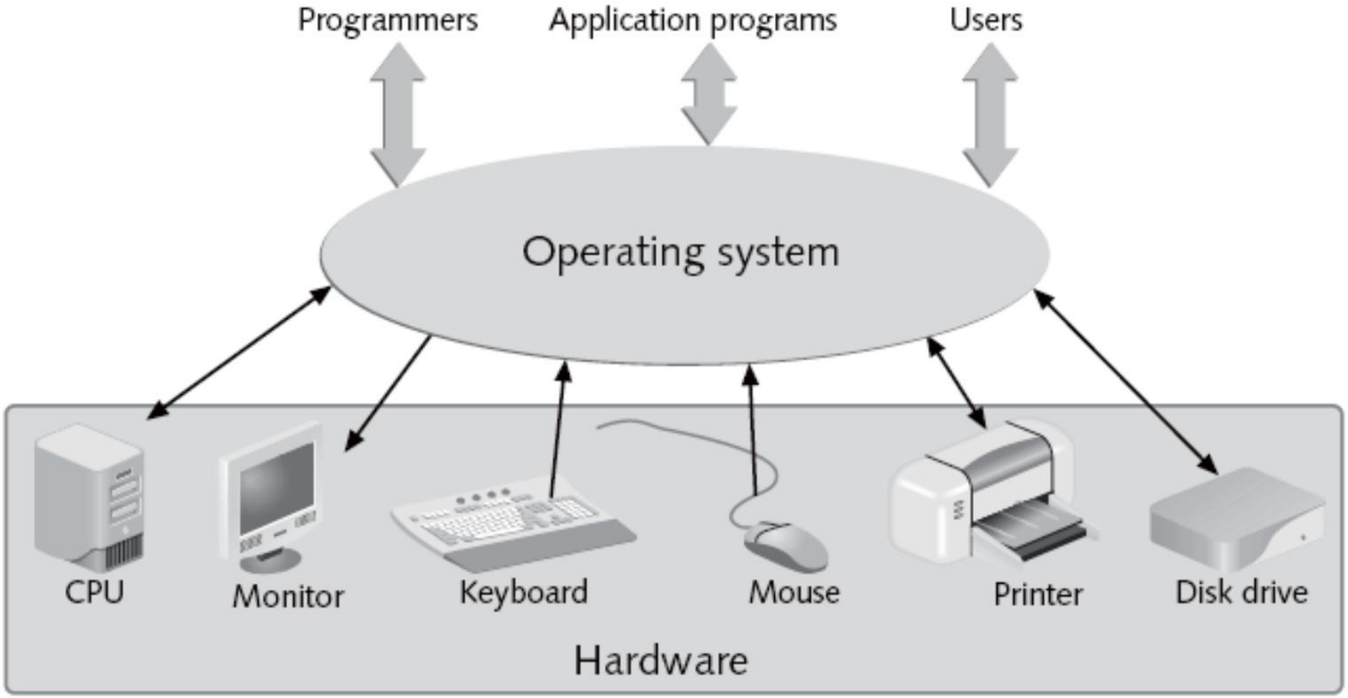


Figure 1-1 Operating system model

PC Operating Systems

- A **personal computer** system, or **PC**, is usually a stand-alone machine
  - E.g., desktop or laptop computer
- A PC OS conducts all the input, output, processing, and storage operations on a single computer

Figure 1-2 Common PC operating systems

Server Operating Systems and Networks

- A computer **network** lets PCs share resources
- A **server OS** controls the operations of a **server** or **host**, which accepts requests from **clients**
- **Peer-to peer networks** are an alternative to server-based networks
  - Each system on the network is both a server and client

Server Operating Systems and Networks (continued)

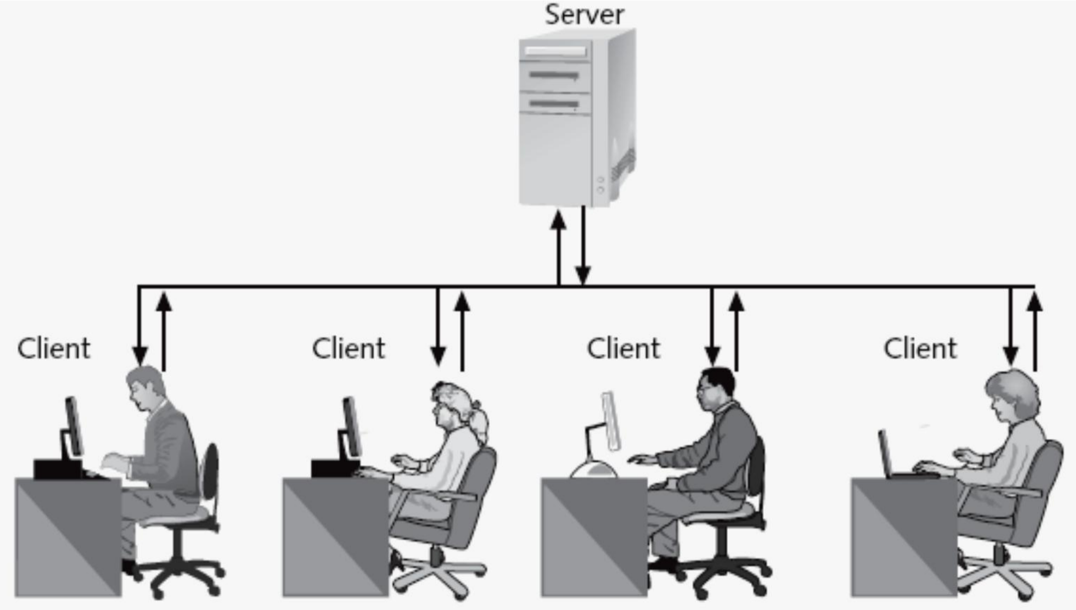


Figure 1-3 Relationship of a server and clients on a network

Introducing the UNIX and Linux Operating Systems

- UNIX/Linux is used on systems functioning as:
  - Servers, clients, client/server workstations, and stand-alone workstations
- UNIX/Linux are **multiuser**/**multitasking** systems
- Some characteristics of UNIX/Linux systems:
  - Portability
  - Stable, reliable, and versatile
  - Thousands of applications are written for them
  - Many security options
  - Well suited for networked environments

A Brief History of UNIX

- Originally developed at AT&T Bell Labs
  - Late 1960s and early 1970s
  - Distributed in source code form
- Two standard versions evolved:
  - AT&T Bell Labs produced **SystemV (SysV)**
  - UC Berkeley developed **BSD**
- Linux might be considered a more integrated version of UNIX than its predecessors
- **POSIX**: effort of experts from industry, academia, and government to standardize UNIX

UNIX Concepts

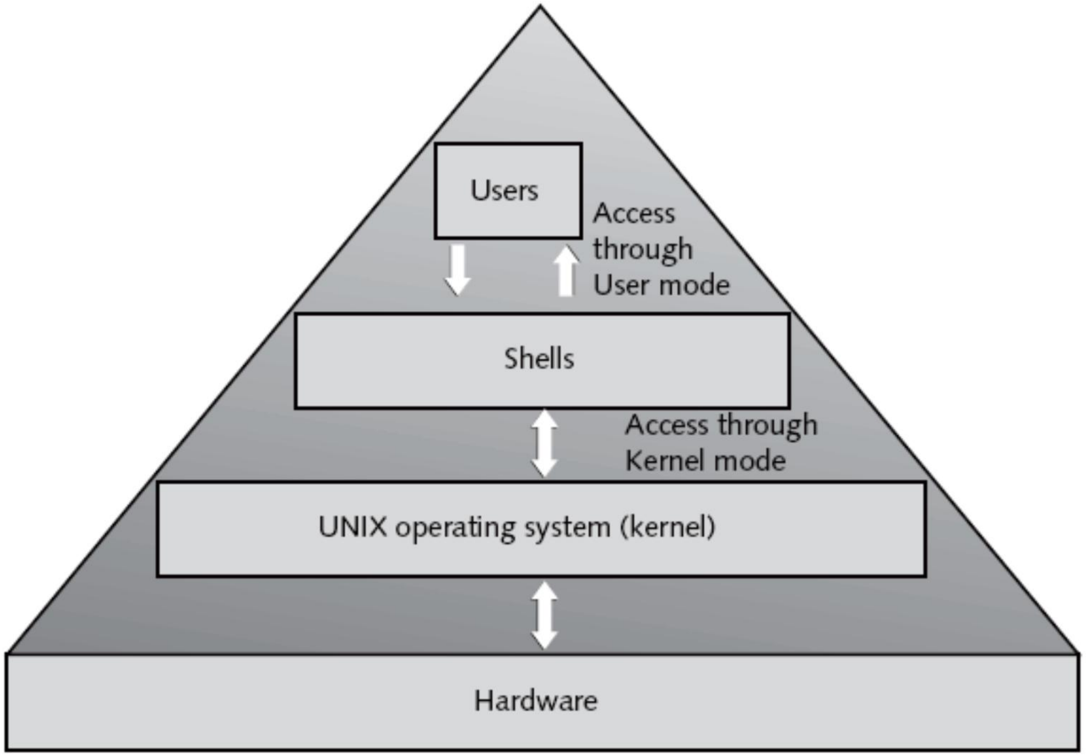


Figure 1-4 Layers of a UNIX system

Linux and UNIX

- Linux is a UNIX-like operating system
  - Not written from the traditional UNIX code
  - Kernel created to look and act like UNIX
    - Enhancements include the POSIX standards
    - Linus Torvalds released it free of charge in 1991
  - Many distributions are available:
    - Debian GNU/Linux
    - Fedora
    - Red Hat Enterprise Linux
    - openSUSE Linux
    - Ubuntu

Introducing UNIX/Linux Shells

- Shell: program that interprets commands you enter from keyboard
  - **Bourne shell,** developed by S. Bourne (AT&T Bell Labs), was the first UNIX command processor
  - Another Bell employee developed the **Korn shell**
    - History feature
  - **C shell** is designed for C programmers' use
  - Linux uses **Bash shell** as its default shell
- **Graphical user interface (GUI)** desktop can open a **terminal window**

Introducing UNIX/Linux Shells (continued)

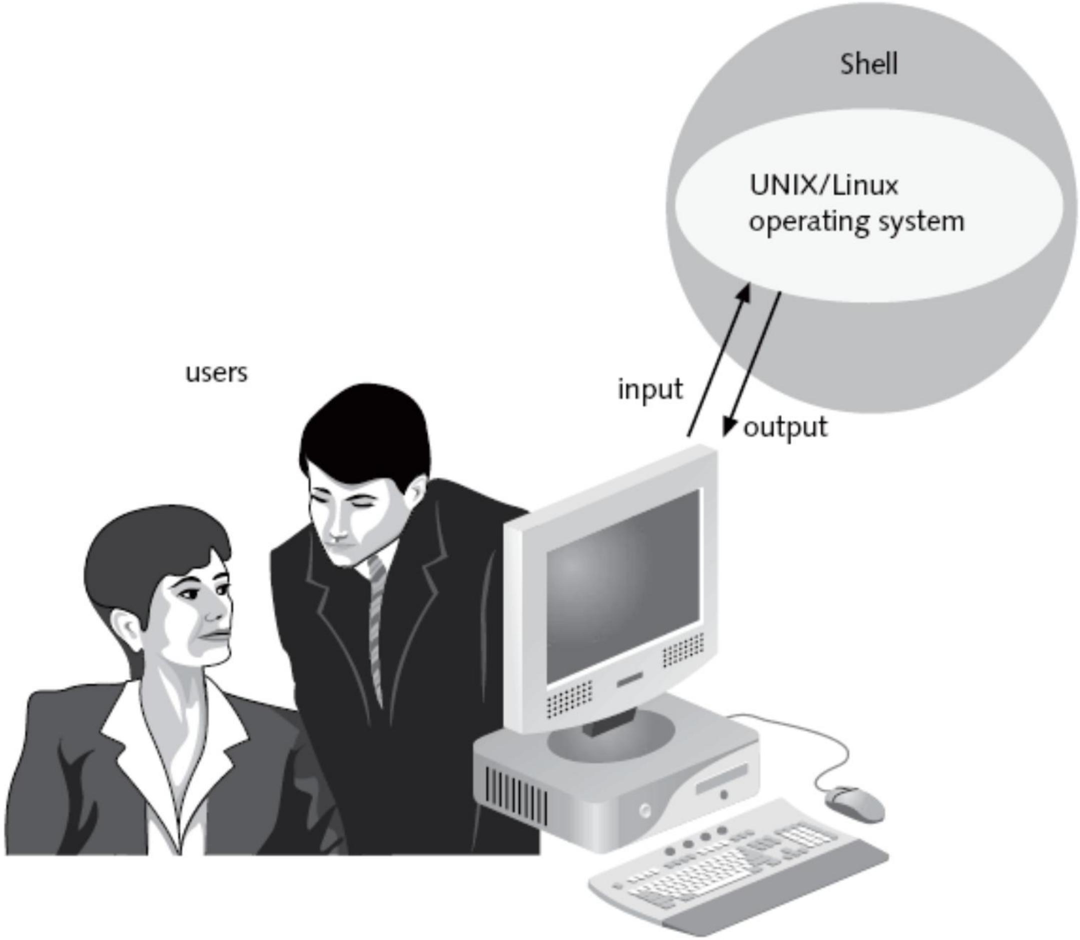


Figure 1-5 Shell's relationship to the user, operating system, and computer

Choosing your Shell

- Shells do much more than interpret commands
  - Extensive built-in commands turn shells into first-class programming languages
- A default shell is associated with your account when it is created
  - You may switch to another shell after you log in
- Many users prefer the Bash shell
- Other shells:
  - Bourne, ksh, csh, ash, tcsh, zsh

Switching from Shell to Shell

- Switch to another shell by typing the shell’s name on the command line
  - For example, type *tcsh*, *bash*, or *ash*
  - Work in that shell until you:
    - Log in again
    - Type another shell name on the command line
- Users often use one shell for writing shell scripts and another for interacting with a program

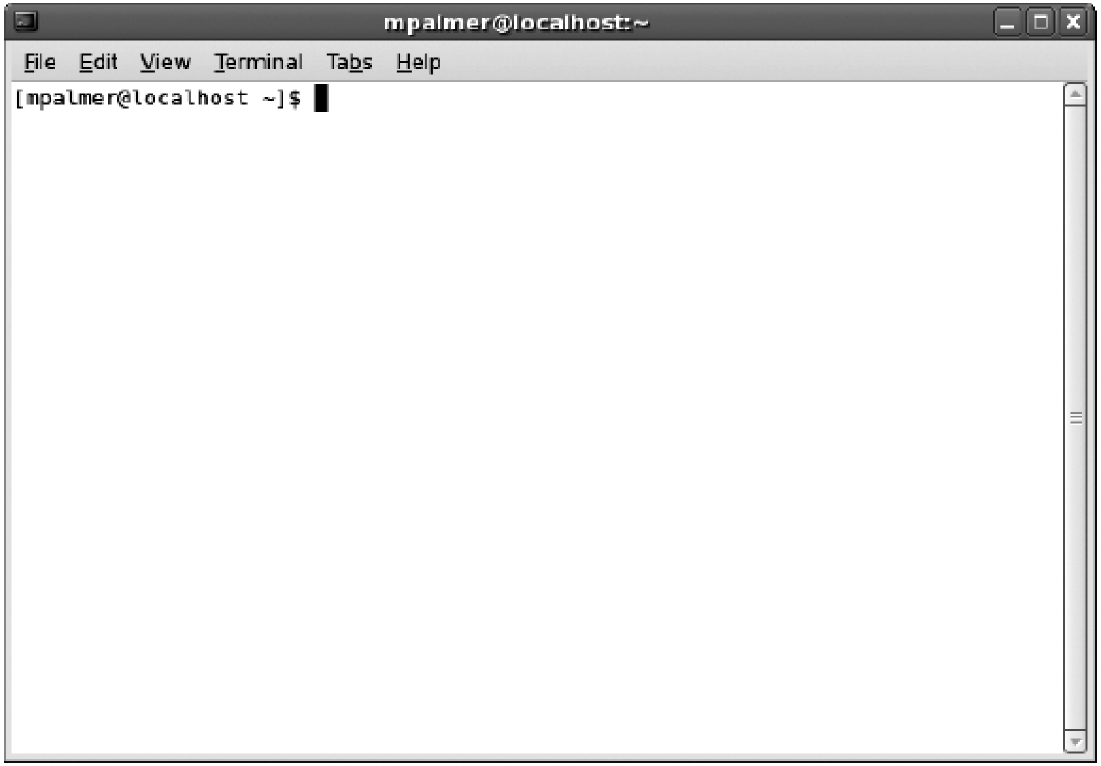
Choosing User Names and Passwords

- Log in using a unique user name and password
  - User name is the same name used for electronic mail
    - Some UNIX versions recognize only first 8 characters
    - Most versions of Linux recognize up to 32 characters
  - Must choose a password
    - Must have 6+ characters in newer versions
    - Must be hard to guess!
    - Change your password using *passwd*
- Common ways to access UNIX/Linux systems:
  - Telnet, SSH, client SW, dumb terminal, etc.

Connecting to UNIX/Linux Using Telnet or SSH

- Telnet is a terminal emulation program
  - Example: *telnet lunar.campus.edu*
- Computers in a network are identified by **IP address** and (sometimes) a **domain name**
  - Examples: *172.16.1.61*, *research.campus.edu*
- **Secure Shell (SSH)** was developed for UNIX/Linux systems to provide authentication for TCP/IP applications
  - Example: *ssh user@hostname*

Logging In to UNIX/Linux



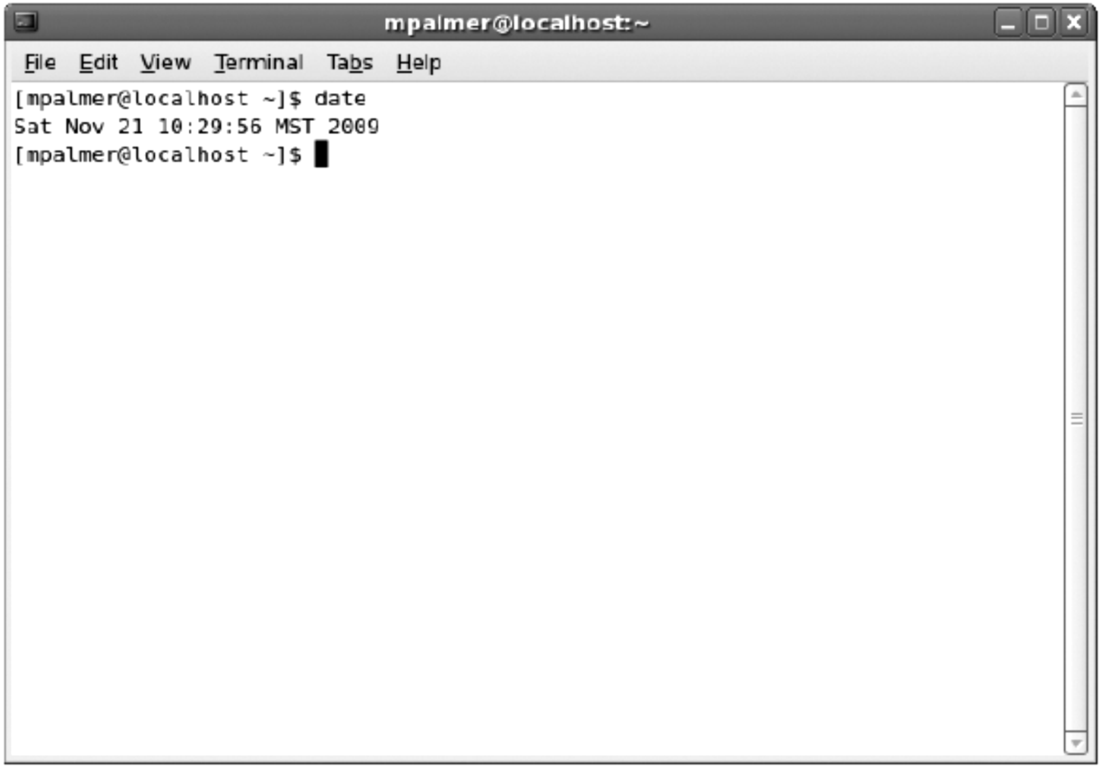
```
[mpalmer@localhost ~]$
```

Figure 1-6 Terminal window in Fedora

Using Commands

- To interact with UNIX/Linux, you enter a **command**
  - UNIX/Linux are **case sensitive**
    - *John* differs from *john*
  - Two categories:
    - User-level commands
    - System-administration commands
  - Must know a command’s **syntax** to enter it properly
    - Need to know **options** and **arguments**
  - Commands are typed on the **command line**

The date Command

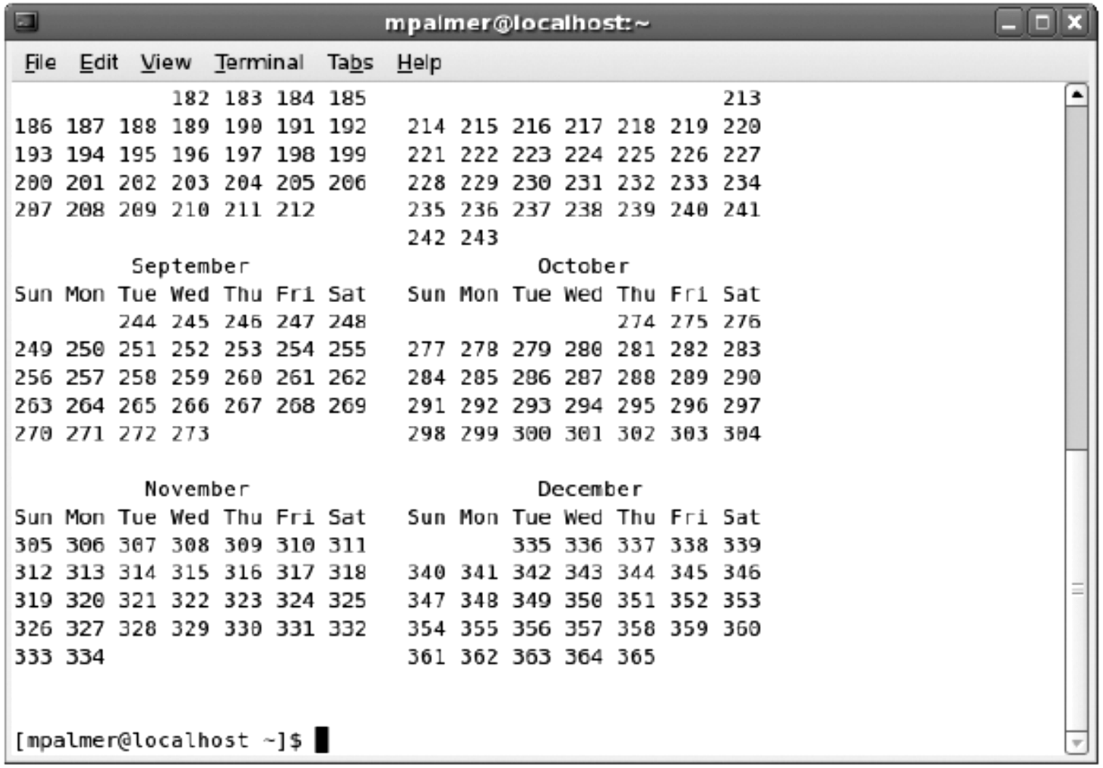


```
[mpalmer@localhost ~]$ date
Sat Nov 21 10:29:56 MST 2009
[mpalmer@localhost ~]$ 
```

Figure 1-7 Using the *date* command

-u option displays the time in Greenwich Mean Time (GMT)

The cal Command



```
            182 183 184 185                           213
186 187 188 189 190 191 192   214 215 216 217 218 219 220
193 194 195 196 197 198 199   221 222 223 224 225 226 227
200 201 202 203 204 205 206   228 229 230 231 232 233 234
207 208 209 210 211 212       235 236 237 238 239 240 241
                              242 243
         September                      October
Sun Mon Tue Wed Thu Fri Sat   Sun Mon Tue Wed Thu Fri Sat
        244 245 246 247 248                   274 275 276
249 250 251 252 253 254 255   277 278 279 280 281 282 283
256 257 258 259 260 261 262   284 285 286 287 288 289 290
263 264 265 266 267 268 269   291 292 293 294 295 296 297
270 271 272 273               298 299 300 301 302 303 304

          November                      December
Sun Mon Tue Wed Thu Fri Sat   Sun Mon Tue Wed Thu Fri Sat
305 306 307 308 309 310 311           335 336 337 338 339
312 313 314 315 316 317 318   340 341 342 343 344 345 346
319 320 321 322 323 324 325   347 348 349 350 351 352 353
326 327 328 329 330 331 332   354 355 356 357 358 359 360
333 334                       361 362 363 364 365


[mpalmer@localhost ~]$ 
```

Figure 1-8 Using the *cal* command to determine the Julian date

-j option used to determine the Julian date

The who Command

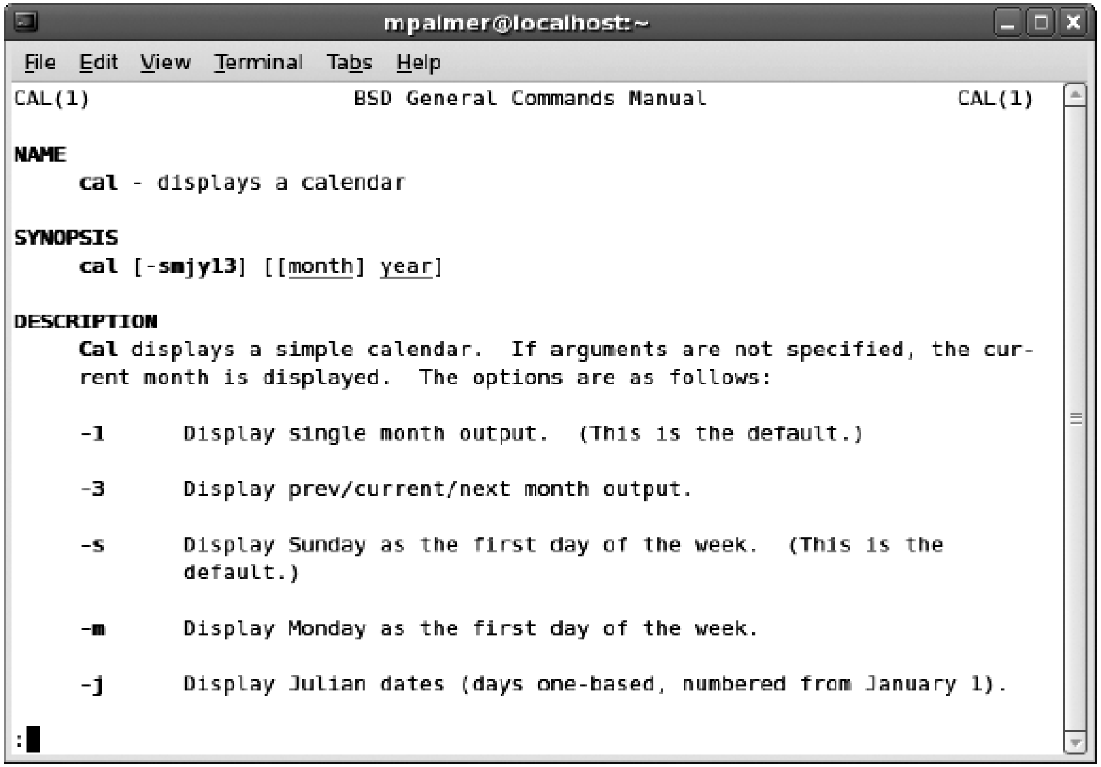
- Determines information about who is logged in
  - Important for the administrator
  - Commonly used options include:
    - *am I* for information about your session
    - *whoami* to see what account you are using
    - *-H* to show column headings
    - *-u* to show idle time for each user
    - *-q* for a quick list and total of users logged in
    - *-b* to verify when the system was last booted

The clear Command

- As you continue to enter commands, your screen might become cluttered
- Use the *clear* command to clear your screen
  - No options or arguments

The man Program

- Online manual called the **man pages**

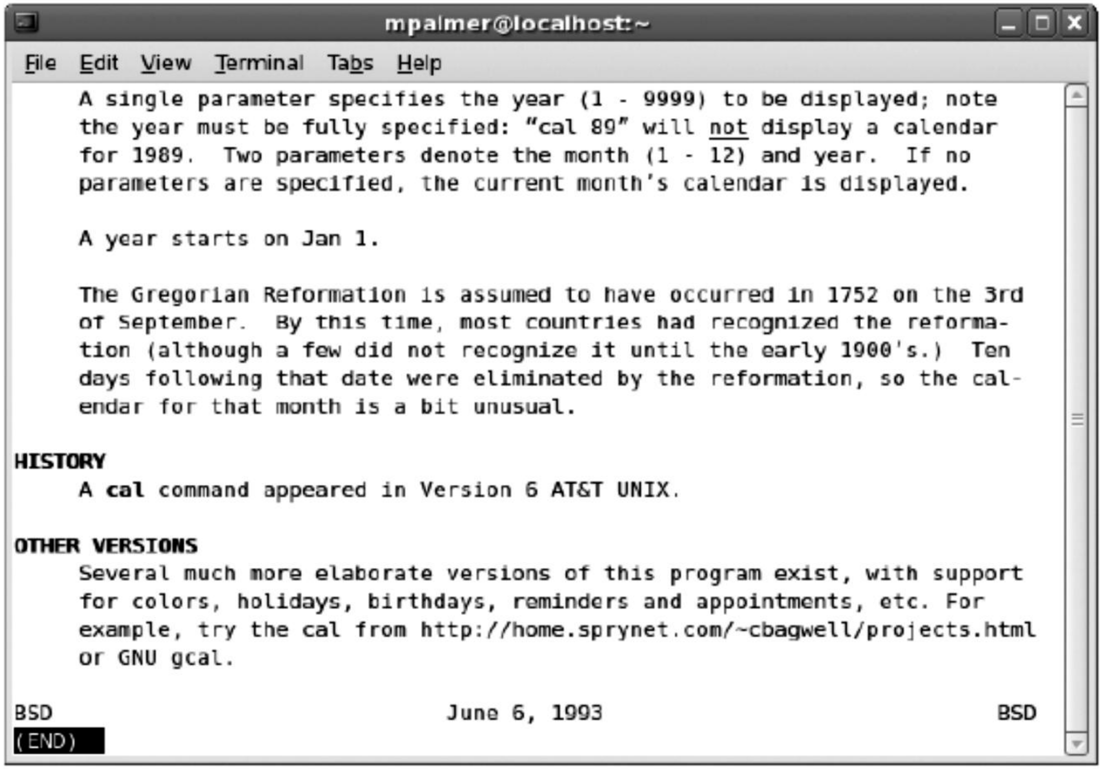


```
CAL(1)                    BSD General Commands Manual                   CAL(1)

NAME
     cal - displays a calendar

SYNOPSIS
     cal [-smjy13] [[month] year]

DESCRIPTION
     Cal displays a simple calendar.  If arguments are not specified, the cur-
     rent month is displayed.  The options are as follows:

     -1      Display single month output.  (This is the default.)

     -3      Display prev/current/next month output.

     -s      Display Sunday as the first day of the week.  (This is the
             default.)

     -m      Display Monday as the first day of the week.

     -j      Display Julian dates (days one-based, numbered from January 1).

:
```

Figure 1-9 man page for the *cal* command

The man Program (continued)

- History section shows that command appeared in Version 6 AT&T UNIX

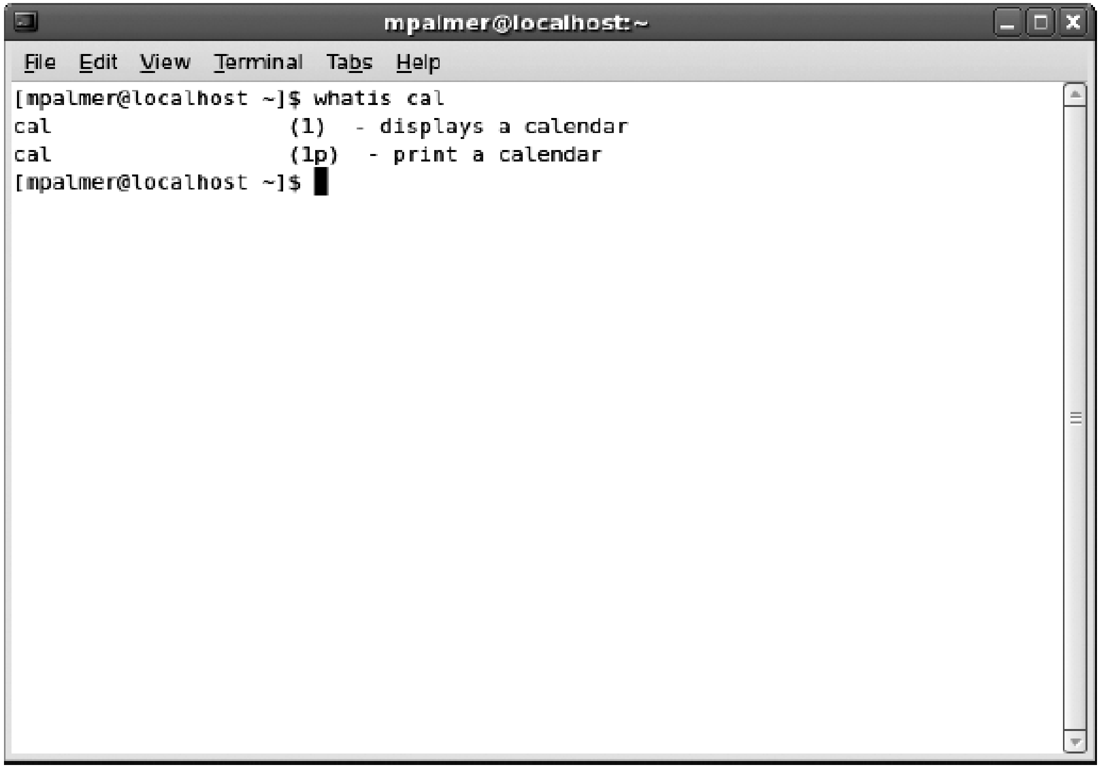


```
     A single parameter specifies the year (1 - 9999) to be displayed; note
     the year must be fully specified: "cal 89" will not display a calendar
     for 1989.  Two parameters denote the month (1 - 12) and year.  If no
     parameters are specified, the current month's calendar is displayed.

     A year starts on Jan 1.

     The Gregorian Reformation is assumed to have occurred in 1752 on the 3rd
     of September.  By this time, most countries had recognized the reforma-
     tion (although a few did not recognize it until the early 1900's.)  Ten
     days following that date were eliminated by the reformation, so the cal-
     endar for that month is a bit unusual.

HISTORY
     A cal command appeared in Version 6 AT&T UNIX.

OTHER VERSIONS
     Several much more elaborate versions of this program exist, with support
     for colors, holidays, birthdays, reminders and appointments, etc. For
     example, try the cal from http://home.sprynet.com/~cbagwell/projects.html
     or GNU gcal.

BSD                              June 6, 1993                              BSD
(END)
```

Figure 1-10 Additional information from the man documentation for the *cal* command

The whatis Command

- Administrator may need to execute *whatis* to create database first



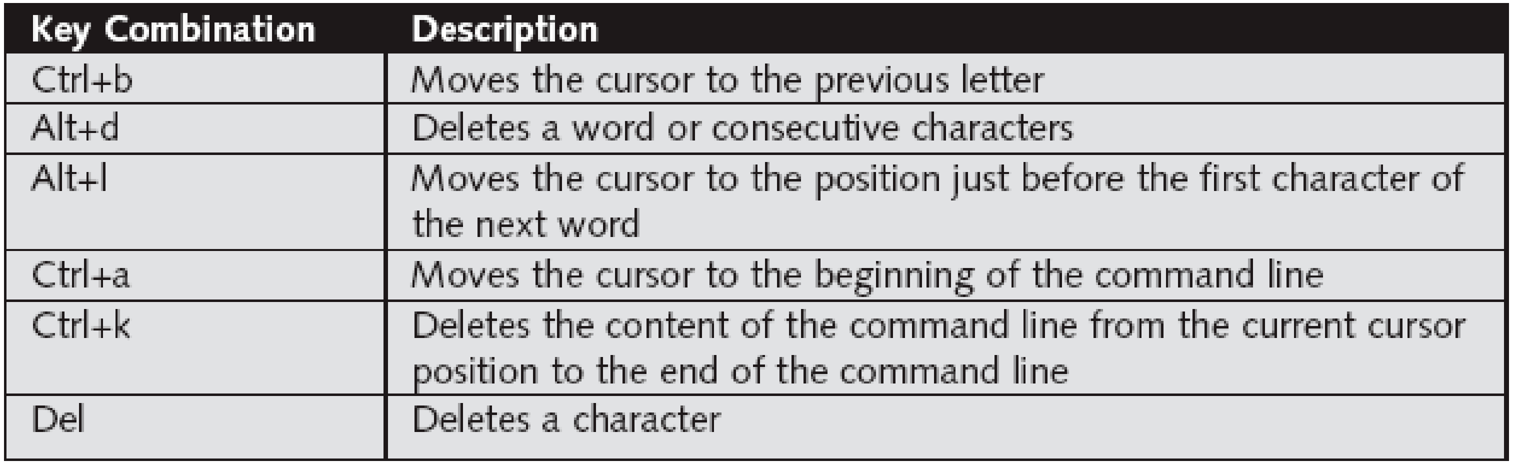
```
[mpalmer@localhost ~]$ whatis cal
cal                  (1)  - displays a calendar
cal                  (1p)  - print a calendar
[mpalmer@localhost ~]$
```

Figure 1-11 Using *whatis* for a quick summary of the *cal* command

Command-line Editing

- Shells support certain keystrokes for performing command-line editing
  - Bash supports ← and → to move cursor

Table 1-1 Common Alt, Ctrl, and Del key combinations for command-line editing



Key Combination	Description
Ctrl+b	Moves the cursor to the previous letter
Alt+d	Deletes a word or consecutive characters
Alt+l	Moves the cursor to the position just before the first character of the next word
Ctrl+a	Moves the cursor to the beginning of the command line
Ctrl+k	Deletes the content of the command line from the current cursor position to the end of the command line
Del	Deletes a character

Multiple Command Entries

- Type multiple commands on command line by separating them with a semicolon
  - *date ; cal*

The Command-line History

- Most shells keep a list of recently used commands
  - You can recall a command without retyping it
    - Access command history with up/down arrow keys
    - Press Enter to execute command once you find it
  - Feature saves time and decreases frustration

Logging Out of UNIX/Linux

- When you are done, log out for security
  - Ends your current process
  - Indicates to OS you are finished
  - For the Bourne, Korn, or Bash shells:
    - Enter *exit* on command line
    - Or, press Ctrl+d
  - In C shell, enter *logout* on the command line
- However, if you are using a GUI, these commands will only close terminal window
  - Use the Log Out option for the desktop instead

Understanding the Role of the UNIX/Linux System Administrator

- A **system administrator** manages the system
  - Also called the **superuser**
    - Adds new users
    - Deletes old accounts
    - Ensures that system performs services well and efficiently for all users
  - Unique user name: **root**
- **Ordinary users** are all other users

The System Administrator’s Command Line

- Default setting: `[root@`*`hostname`*` root]#`
  - – *hostname:* name of computer the system administrator logged in to
    - May simply be localhost: refers to the local computer

The Ordinary User’s Command Line

- Common formats：

```
[username@hostname username] $
[username@hostname ~] $
username@hostname: →
```

  - *username:* user’s login name
  - *hostname:* name of computer to which user is logged in
  - Note: ~ refers to the user’s home directory

Changing Passwords

- A password is confidential and secures your work on the system
- To change your password, use *passwd*
  - Some rules will apply depending on system
  - Administrators can add rules of their own
  - You must know your current password to change it
  - If account does not have a password, use *passwd* command to create one

Viewing Files Using the cat, more, less, head, and tail Commands

- *more* and *less* display a file one screen at a time
  - *more* scrolls only down
  - *less* enables you to scroll down and up
- *cat* displays the whole file at one time
  - Comes from “concatenate”: to link
- Use *head* or *tail* to view first or last lines of a file
  - 10 lines by default

Redirecting Output

- > is an **output redirection operator**
  - Creates a new file or overwrites an existing file by attaching it to a command that produces output
  - Examples:

    who > current_users

    cat > filename
- To append output to an existing file, use >>
  - Adds information to the end of an existing file without overwriting that file

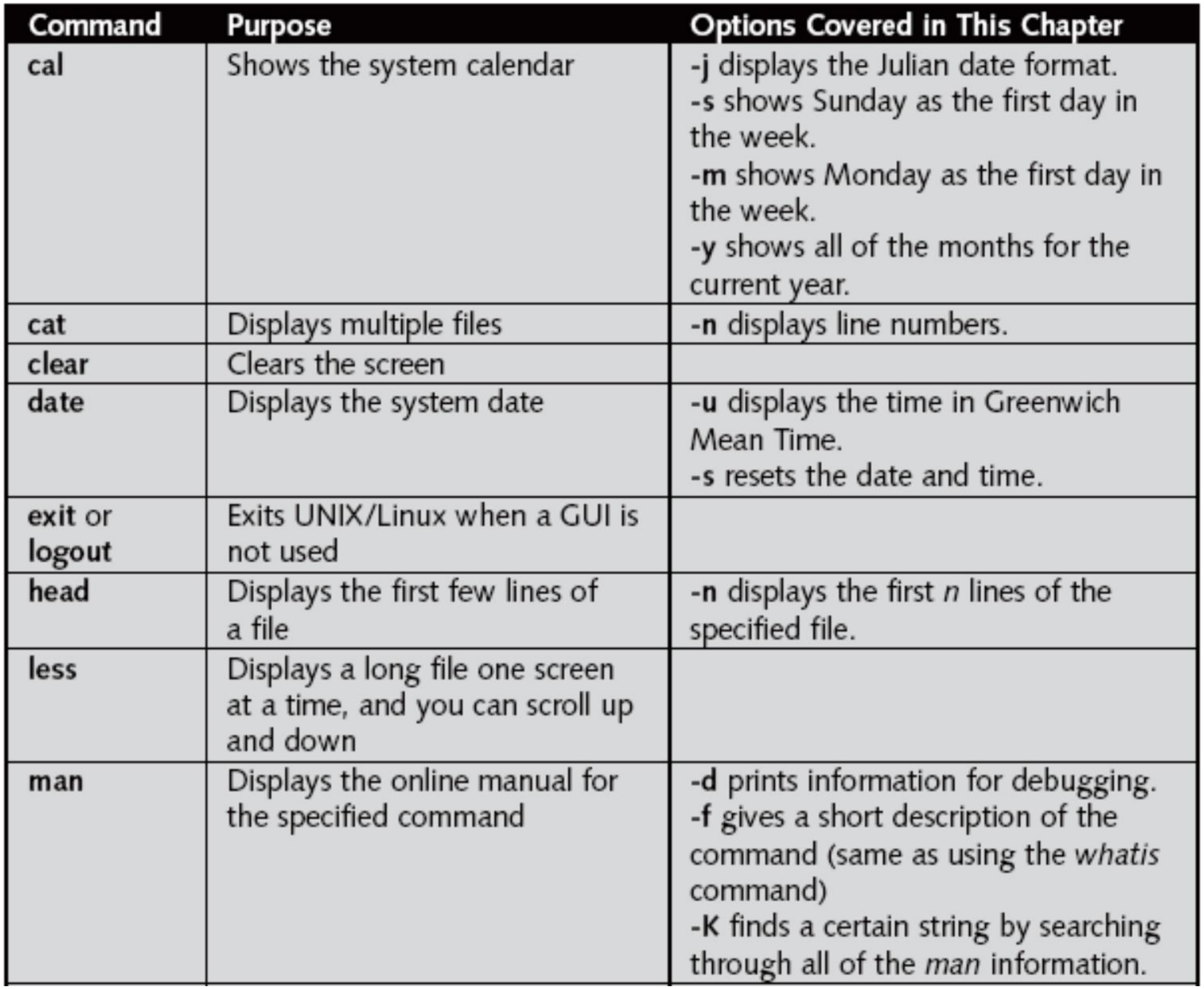
Summary

- The OS is the most fundamental computer program
- UNIX/Linux OSs are multiuser and multitasking systems
- UNIX/Linux systems can be configured as servers, client workstations, client/server workstations, or stand-alone workstations
- Concept of OS layered components originated with UNIX
- In UNIX/Linux, you communicate with OS programs through an interpreter called the shell

Summary (continued)

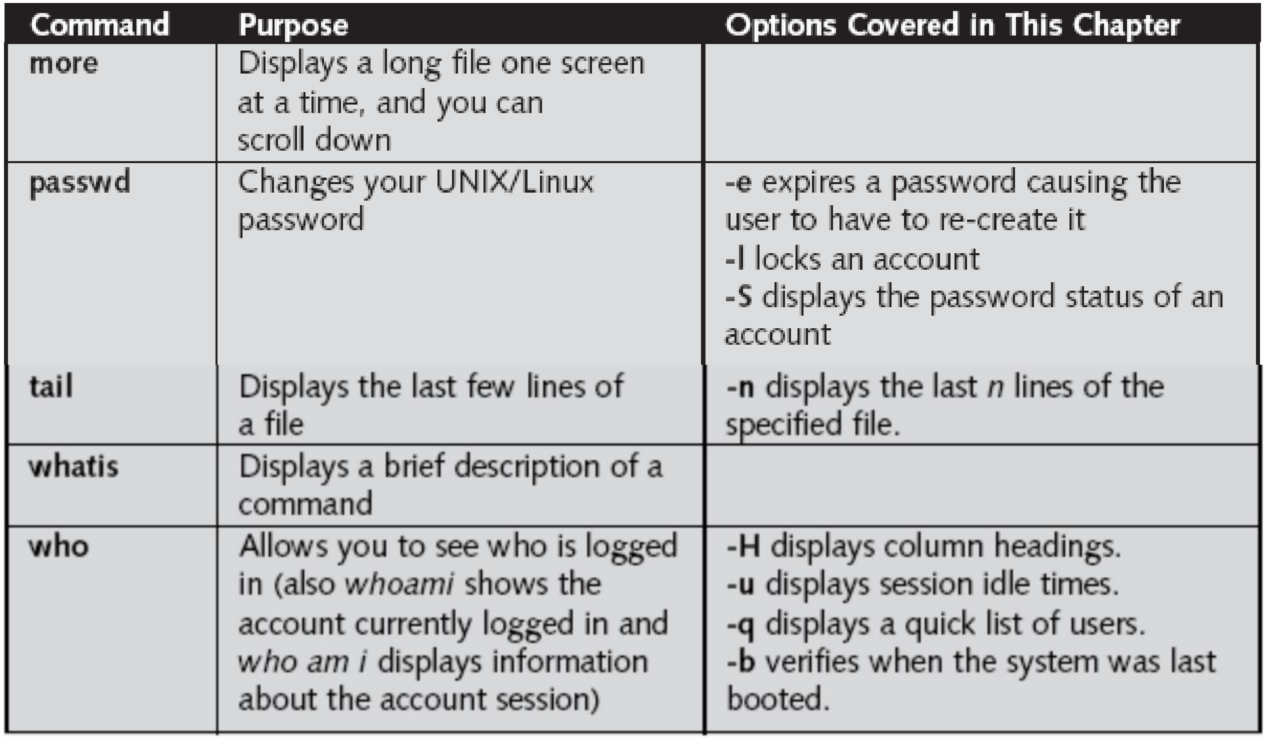
- In UNIX/Linux, the system administrator sets up accounts for ordinary users
- The commands you type to work with UNIX/Linux have a strict syntax
  - Learn syntax by referring to the man pages
  - Examples of commands: who, cal, date, passwd
- Shells provide command-line editing capabilities and keep a history of your recently used commands
- Use cat, less, more, head, and tail to view files

Command Summary



Command	Purpose	Options Covered in This Chapter
cal	Shows the system calendar	**-j** displays the Julian date format. **-s** shows Sunday as the first day in the week. **-m** shows Monday as the first day in the week. **-y** shows all of the months for the current year.
cat	Displays multiple files	**-n** displays line numbers.
clear	Clears the screen	
date	Displays the system date	**-u** displays the time in Greenwich Mean Time. **-s** resets the date and time.
exit or **logout**	Exits UNIX/Linux when a GUI is not used	
head	Displays the first few lines of a file	**-n** displays the first *n* lines of the specified file.
less	Displays a long file one screen at a time, and you can scroll up and down	
man	Displays the online manual for the specified command	**-d** prints information for debugging. **-f** gives a short description of the command (same as using the *whatis* command) **-K** finds a certain string by searching through all of the *man* information.

Command Summary (continued)



Command	Purpose	Options Covered in This Chapter
more	Displays a long file one screen at a time, and you can scroll down	
passwd	Changes your UNIX/Linux password	**-e** expires a password causing the user to have to re-create it -l locks an account **-S** displays the password status of an account
tail	Displays the last few lines of a file	**-n** displays the last *n* lines of the specified file.
whatis	Displays a brief description of a command	
who	Allows you to see who is logged in (also *whoami* shows the account currently logged in and *who am i* displays information about the account session)	**-H** displays column headings. **-u** displays session idle times. **-q** displays a quick list of users. **-b** verifies when the system was last booted.