1

Answers to Even-numbered Exercises

2. Why is Linux popular? Why is it popular in academia?

   Linux is portable, is based on standards, is written in C, has a kernel programming interface, can support many users, and can run multiple tasks simultaneously. For more information refer to “What Is So Good About Linux?” on page 8.

   The source code for the operating system is readily available so students can understand more easily how Linux works and can modify the code further to understand its operation and to change the way it works. For more information refer to “The Code Is Free” on page 5.

4. What is Linux? What is the Free Software Foundation/GNU? Which parts of the Linux operating system did each provide? Who else has helped build and refine this operating system?

   Linux is the name of the operating system kernel developed by Linus Torvalds, which has since been expanded and improved by thousands of people on the Internet.

   The Free Software Foundation (www.fsf.org) is the principal organizational sponsor of the GNU Project. GNU developed many of the tools, including the C compiler, that are part of the Linux operating system.

   Torvalds’ kernel and GNU’s tools work together as the Linux operating system.

6. What is a distribution? What does it contain? Name three distributions.

   A distribution typically includes word processors, spreadsheets, media players, database applications, and a program to install the distribution. In addition, a distribution includes libraries and utilities from the GNU Project and graphics support from the X Window System.

   All distributions are based on the same upstream code, although each might include different applications and tools. Distributions distinguish themselves in the areas of package management and installation tools, policies, community, and support.

   Distributions include Fedora/Red Hat Enterprise Linux, Ubuntu, Mandriva, openSUSE, Debian, Gentoo, and Mageia.

8. What is a utility program?

   A utility (program), sometimes referred to as a command, performs a task that is frequently related to the operating system. A utility is simpler than an application program, although no clear line separates the two. Linux distributions include many utilities. You can also download many utilities from the Internet.

   Examples of utilities are `cp` (copies a file), `ls` (lists information about files), `ssh` (securely connects to a remote computer), and `df` (lists information about free space on system devices such as hard disks).

10. How can you use utility programs and a shell to create your own applications?

    You can write a shell script, also called a shell program, or a batch file under DOS. A shell script is one or more command lines contained in a file. Make the file executable and give the name of the file as a command. The shell then executes the commands in the file as though you had typed each command individually. (You might need to give the *command* as *./command*.)

12. What is the difference between a multiuser and a multitasking system?

    A multiuser system can support more than one user at a time.

    A multitasking system can process more than one task at a time.

14. Approximately how many people wrote Linux? Why is this project unique?

    Many thousands of people have contributed to the Linux operating system using the Internet. This project is unique because a project of this magnitude, using free software, had never been attempted before.

2

Answers to Even-numbered Exercises

2. What is an installer? What is the name of the Fedora/RHEL installer?

   The installer is a tool that automates the process of installing Linux and makes the installation process easier and friendlier. Fedora/RHEL uses the Anaconda installer.

4. A system boots from the hard disk. To install Linux, you need it to boot from a DVD. How can you make the system boot from a DVD?

   As the system boots, go into the BIOS setup and change the order of the devices the system tries to boot from. Revise the order so that the system first tries to boot from the DVD and then tries to boot from the hard disk.

6. What is an ISO image? How do you burn an ISO image to a DVD?

   An ISO image is an exact copy of what is on a DVD. When you burn an ISO image to a DVD, you must use a special command that is part of most DVD-writing software; you cannot copy an ISO image to a DVD the same way you copy other files. The special command has a label similar to **Record CD from CD Image** or **Burn CD Image**.

8. What are RAM disks? How are they used during installation?

   A RAM disk is random access (system) memory that is made to look like a hard disk. Tools used during the installation process are copied to RAM disks. RAM disks allow the installation process to run through the specification and design phases without writing to the hard disk. Thus RAM disks enable you to quit installing the system and, unless the installer initialized the hard disk, leave the hard disk as it was at any point before the system warns it is going to write to the hard disk.

3

Answers to Even-numbered Exercises

2. Describe the Anaconda installer.

   Anaconda is written in Python and C. It identifies the hardware present in the system, builds the necessary filesystems, and installs the Fedora/RHEL operating system. Anaconda can run in graphical interactive mode or in automated mode (Kickstart).

4. Why is it important to test the installation medium? How can you do so?

   It is important to verify the integrity of a downloaded image to ensure that it will be functional and that it has not been tampered with.

   You can test the installation medium by selecting the **Test this media** in the Boot menu (Figure 3-4), by clicking **Verify** in the Software/Installation Source screen during installation (page 74), or by using manually using `sha256sum` before installation (page 53).

6. When might you specify an **ext2** filesystem instead of **ext4**?

   Use **ext2** for partitions whose data does not change often, such as **/boot**. The added overhead of the **ext4** journal offers no benefit on these filesystems.

8. What do you need to do before you can install Fedora as the second operating system on a Windows machine (to create a dual-boot system)?

   You need to back up important data and create free space on the disk to install Fedora. You can create free space by deleting or shrinking partitions.

10. How would you turn off DMA (direct memory access) for all disk controllers when you install a new system?

You need to specify the **nodma** boot parameter as you boot the system. To specify a boot parameter, you must interrupt the automatic boot process by pressing the SPACE bar while the system is counting down when you first boot the system. When you press the SPACE bar, Fedora displays the Fedora Boot menu. Use the ARROW keys to highlight the selection you want before proceeding. With the desired selection highlighted, press the TAB key to display the boot command-line parameters. Enter a SPACE followed by **nodma** and press RETURN to boot the system.

4

Answers to Even-numbered Exercises

2. Give three examples of poor password choices. What is wrong with each?

   Examples of poor password choices follow:

finger	word in the dictionary
tom	username
aqbfgya	does not contain a number
5q	too short

4. What is a context menu? How does a context menu differ from other menus?

   A context menu has choices that apply specifically to the window or object you click and that differ from window to window and from object to object. Some windows do not have context menus. Frequently a right-click displays a context menu.

6. How would you swap the effects of the right and left buttons on a mouse? What is the double-click speed? How would you change it?

   The Mouse & Touchpad window enables you to change a mouse from right-handed to left-handed, and vice versa. The double-click speed specifies how quickly you must click a mouse button before the system considers the action to be a double-click and not two single clicks. You can change this characteristic by using the Mouse & Touchpad window.

8. What is Nautilus? What does it allow you to do?

   Nautilus is the GNOME file manager. You can use it to copy, move, open, and execute files.

10. What are the functions of a Window Operations menu? How do you display this menu?

    Right-clicking the window titlebar displays the Window Operations menu. This menu allows you to move, resize, close, and otherwise manipulate a window.

12. When you are working on the command line, how do you erase (back up over) a character? How do you delete the line you are typing? How do you terminate a program?

    You use the erase key, usually BACKSPACE, to remove characters you have just typed. The line kill key, usually CONTROL-U or CONTROL-X, deletes the line you are entering. Press the interrupt key, usually CONTROL-C, to abort the program that is running.

14. How does the mouse pointer change when you move it to the edge of a window? What happens when you left-click and drag the mouse pointer when it looks like this? Repeat this experiment with the mouse pointer at the corner of a window.

    The mouse pointer changes to an arrow pointing to a line. When you drag this arrow, you resize the window. When you position the pointer on an edge of the window, you can resize the window in one direction. When you position the pointer on a corner, you can resize in both directions at once.

16. What happens when you run nano from the Enter a Command window? Where does the output go?

    When you run nano in this manner, the output is lost.

18. How many man pages are in the Devices subsection of the system manual? (*Hint:* Devices is a subsection of Special Files.)

    Approximately 60. The following command shows exactly how many man pages are in the Devices subsection of the system manual:

    ```
    $ ls /usr/share/man/man4 | wc -l
    ```

5

Answers to Even-numbered Exercises

2. Using sort as a filter, rewrite the following sequence of commands:

```
$ sort list > temp
$ lpr temp
$ rm temp
```

```
$ cat list | sort | lpr
```

4. Assume the following files are in the working directory:

```
$ ls
intro      notesb     ref2       section1   section3   section4b
notesa     ref1       ref3       section2   section4a  sentrev
```

Give commands for each of the following, using wildcards to express filenames with as few characters as possible.

a. List all files that begin with **section**.

```
$ ls section*
```

b. List the **section1**, **section2**, and **section3** files only.

```
$ ls section[1-3]
```

or

```
$ ls section[123]
```

c. List the **intro** file only.

```
$ ls i*
```

d. List the **section1**, **section3**, **ref1**, and **ref3** files.

```
$ ls *[13]
```

6. Give a command to

   a. Redirect standard output from a sort command to a file named **phone_list**. Assume the input file is named **numbers**.

   ```
   $ sort numbers > phone_list
   ```

   b. Translate all occurrences of the characters **[** and **{** to the character **(**, and all occurrences of the characters **]** and **}** to the character **),** in the file **permdemos.c**. (*Hint:* Refer to the tr man page.)

   ```
   $ cat permdemos.c | tr '[{}]' '(())' or
   $ tr '[{}]' '(())' < permdemos.c
   ```

   c. Create a file named **book** that contains the contents of two other files: **part1** and **part2**.

   ```
   $ cat part[12] > book
   ```

8. Give an example of a command that uses grep

   a. With both input and output redirected.

   ```
   $ grep \$Id < *.c > id_list
   ```

   b. With only input redirected.

   ```
   $ grep -i suzi < addresses
   ```

   c. With only output redirected.

   ```
   $ grep -il memo *.txt > memoranda_files
   ```

   d. Within a pipeline.

   ```
   $ file /usr/bin/* | grep "Again shell script" | sort -r
   ```

   In which of the preceding cases is grep used as a filter?

   Part **d** uses grep as a filter.

10. When you use the redirect output symbol (>) on a command line, the shell creates the output file immediately, before the command is executed. Demonstrate that this is true.

```
$ ls aaa
ls: aaa: No such file or directory
$ ls xxxxx > aaa
ls: xxxxx: No such file or directory
$ ls aaa
aaa
```

The first command shows the file **aaa** does not exist in the working directory. The second command uses ls to attempt to list a nonexistent file (**xxxxx**) and sends standard output to **aaa**. The ls command fails and sends an error message to standard error (i.e., displays it on the screen). Even though the ls command failed, the empty file named **aaa** exists. Because the ls command failed, it did not create the file; the shell created it before calling ls.

12. Assume permissions on a file allow you to write to the file but not to delete it.

    a. Give a command to empty the file without invoking an editor.

    ```
    $ cat /dev/null > filename
    ```

    b. Explain how you might have permission to modify a file that you cannot delete.

    To delete a file, you must have write and execute permission for the directory holding the file. To write to a file, you must have write permission for the file and execute permission for the parent directory. When you have write permission only for a file and execute permission only for the directory holding the file, you can modify but not delete the file.

14. Why does the **noclobber** variable *not* protect you from overwriting an existing file with cp or mv?

    The **noclobber** variable implements a shell feature that keeps the shell from overwriting a file; it does not work with utilities. Thus it keeps a redirect symbol (>) from allowing the shell to overwrite a file (the shell redirects output) but has no effect when you ask cp or mv to overwrite a file.

16. Create a file named **answer** and give the following command:

    ```
    $ > answers.0102 < answer cat
    ```

    Explain what the command does and why. What is a more conventional way of expressing this command?

    Reading the command line from left to right, it instructs the shell to redirect standard output to **answers.0102**, to redirect standard input to come from **answer**, and to execute the cat utility. More conventionally, the same command is expressed as

    ```
    $ cat answer > answers.0102
    ```

    or simply

    ```
    $ cp answer answers.0102
    ```

6

Answers to Even-numbered Exercises

2. List the commands you can use to perform these operations:

   a. Make your home directory the working directory

   b. Identify the working directory

   a. **cd**; b. **pwd**

4. The df utility displays all mounted filesystems along with information about each. Use the df utility with the **–h** (human-readable) option to answer the following questions.

```
$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda1             1.4G  242M  1.1G  18% /
/dev/hda3              23M   11M   10M  51% /boot
/dev/hda4             1.5G  1.2G  222M  85% /home
/dev/hda7             564M   17M  518M   4% /tmp
/dev/hdc1             984M   92M  842M  10% /gc1
/dev/hdc2              16G   13G  1.9G  87% /gc2
```

   a. How many filesystems are mounted on your Linux system?

   b. Which filesystem stores your home directory?

   c. Assuming your answer to exercise 4a is two or more, attempt to create a hard link to a file on another filesystem. What error message do you get? What happens when you attempt to create a symbolic link to the file instead?

   Following are sample answers to these questions. Your answers will be different because your filesystems are different.

   a. six; b. **/dev/hda4;** c. **ln**: **creating hard link '/tmp/xxx' to 'xxx': Invalid cross-device link**. No problem creating a cross-device symbolic link.

6. You should have read permission for the **/etc/passwd** file. To answer the following questions, use cat or less to display **/etc/passwd**. Look at the fields of information in **/etc/passwd** for the users on the local system.

   a. Which character is used to separate fields in **/etc/passwd**?

   b. How many fields are used to describe each user?

   c. How many users are on the local system?

   d. How many different login shells are in use on your system? (*Hint:* Look at the last field.)

   e. The second field of **/etc/passwd** stores user passwords in encoded form. If the password field contains an **x**, your system uses shadow passwords and stores the encoded passwords elsewhere. Does your system use shadow passwords?

   a. colon (**:**); b. seven; c, d, and e are system dependent

8. Suppose a user belongs to a group that has all permissions on a file named **jobs_list**, but the user, as the owner of the file, has no permissions. Describe which operations, if any, the user/owner can perform on **jobs_list**. Which command can the user/owner give that will grant the user/owner all permissions on the file?

   Initially the user/owner cannot perform any operations involving the file, other than using ls to list it. When the user/owner gives the following command, the user/owner can perform any operation involving the file:

   ```
   $ chmod u+rwx jobs_list
   ```

10. Assume you are given the directory structure shown in Figure 6-2 on page 177 and the following directory permissions:

    ```
    d--x--x---   3 zach pubs 512 2010-03-10 15:16 business
    drwxr-xr-x   2 zach pubs 512 2010-03-10 15:16 business/milk_co
    ```

    For each category of permissions—owner, group, and other—what happens when you run each of the following commands? Assume the working directory is the parent of **correspond** and that the file **cheese_co** is readable by everyone.

    a. **cd correspond/business/milk_co**

    owner: OK; group: OK; other: **Permission denied**

    b. **ls –l correspond/business**

    owner, group, and other: **Permission denied**

    c. **cat correspond/business/cheese_co**

    owner and group: **Is a directory**; other: **Permission denied**

12. What does the **..** entry in a directory point to? What does this entry point to in the root (/) directory?

    The **..** entry is a link to the parent directory. In the case of the root directory, there is no parent, and the **..** entry is a link to the root directory itself.

14. Suppose the working directory contains a single file named **andor**. What error message do you get when you run the following command line?

    ```
    $ mv andor and\/or
    ```

    Under what circumstances is it possible to run the commaìnd without producing an error?

    ```
    $ mv andor and\/or
    mv: cannot move 'andor' to 'and/or': No such file or directory
    $ mkdir and
    $ mv andor and\/or
    $ ls and
    or
    ```

    (The backslash is superfluous.)

16. Do you think the system administrator has access to a program that can decode user passwords? Why or why not? (See exercise 6.)

    Normally, the system administrator cannot decode user passwords. The administrator can assign a new password to a user. Passwords are generally encrypted by a one-way hash so the system can tell when the correct password is entered, but it cannot regenerate the cleartext password. The system applies the hash algorithm to the entered password and checks whether the result matches the stored, encrypted password. A match means the correct password was entered.

    However, if a user has a weak password, the system administrator can use a program such as crack or John the Ripper to decode a password. You can download either of these utilities using yum. The packages are named **crack** and **john**.

18. Explain the error messages displayed in the following sequence of commands:

    ```
    $ ls -l
    total 1
    drwxrwxr-x. 2 max pubs 1024 03-02 17:57 dirtmp
    $ ls dirtmp
    $ rmdir dirtmp
    rmdir: dirtmp: Directory not empty
    $ rm dirtmp/*
    rm: No match.
    ```

    There is a file whose name begins with a period (a hidden file) in the **dirtmp** directory. Use ls with the **–a** option to list the file. Remove the file, and then you will be able to remove the directory.

7

Answers to Even-numbered Exercises

2. Give a command that displays a long listing of the files in **/bin** in reverse chronological order. Give the command again but this time display the output one screen at a time.

```
$ ls -ltr /bin

$ ls -ltr /bin | less
```

4. List the first 20 lines in **/etc/services** that describe TCP ports.

```
$ grep tcp /etc/services | head -20
```

6. What happens when you use diff to compare two binary files that are not identical? (You can use gzip to create the binary files.) Explain why the diff output for binary files is different from the diff output for ASCII files.

   When you use it to compare binary files, diff displays a message saying the files differ when the files differ or no message when the files are the same. The diff utility compares ASCII files on a line-by-line basis; it is not designed to compare binary files on a byte-by-byte basis. Use cmp to compare binary files in that manner.

8. Are any of the utilities discussed in this chapter located in more than one directory on the local system? If so, which ones?

   No. However, some commands that are built into a shell have counterparts that exist as executable files (e.g., echo).

10. Which command can you use to look at the first few lines of a file named **status.report**? Which command can you use to look at the end of the file?

```
$ head status.report
$ tail status.report
```

12. Display a long listing of the files in the **/etc/pam.d** directory hierarchy that are links.

```
$ find /etc/pam.d -type l -exec ls -l {} \;
```

or

```
$ find /etc/pam.d -type l | xargs ls -l
```

14. Display the **/etc/passwd** file, replacing all colons (**:**) with TABS. Display the **/etc/services** file, substituting one SPACE for each occurrence of multiple SPACES.

```
$ cat /etc/passwd | tr ':' '\t'

$ cat /etc/services | tr -s ' '
```

16. Copy **/bin/bash** to the working directory and make two copies so you have three identical files: **bash1**, **bash2**, and **bash3**. Compress **bash1** using gzip and **bash2** using bzip2. Do not change **bash3**. Which utility does the best job of compressing the file? Which does the worst? How big is **bash2.bz2** compared to **bash3**?

```
$ gzip bash1
$ bzip2 bash2
$ ls -l bash*
```

The gzip utility does not do as good a job as bzip2. The **bash2.bz2** file is about 44 percent as big as **bash3**.

18. Try giving these two commands:

```
$ echo cat
$ cat echo
```

Explain the differences between the output of each command.

The first command causes echo to display the characters **c**, **a**, and **t** on the screen. The second command uses cat to copy the contents of a file named **echo** to the screen. If there is no file named **echo**, cat displays an error message.

20. Find or create files that

a. gzip compresses by more than 80 percent.

The gzip utility compresses most text files by more than 80 percent.

b. gzip compresses by less than 10 percent.

The gzip utility compresses most files that are already compressed, such as **jpeg** files, by less than 10 percent.

c. Get larger when compressed with gzip.

The gzip utility expands a file that has already been compressed with gzip. (To compress a gzipped file a second time, you must remove the **.gz** filename extension.)

d. Use **ls –l** to determine the sizes of the files in question. Can you characterize the files in a, b, and c?

Files with repeated information or inefficiently stored information can be compressed the most. Files that have been compressed already store information efficiently and can be compressed only a small amount, not at all, or negatively (expanded).

8

Answers to Even-numbered Exercises

2. Describe two ways to find out who is logged in on some of the other machines attached to your network.

   Use rsh, ssh, or telnet to connect to and run w or who on each host.

   Use finger.

   Log in on the console of each host and run w or who.

4. A software implementation of chess was developed by GNU and is available for free. How can you use the Internet to find and download this program?

   Use a search engine to find **GNU chess** and download the software from an appropriate site. Alternatively, go to the GNU home page and find the page that you can download the software from.

6. If you have access to the World Wide Web, answer the following questions.

   a. Which browser do you use?

      System/user dependent, frequently Chrome or Firefox.

   b. What is the URL of the author of this book's home page? How many links does it have?

      The URL is www.sobell.com; the number of links varies.

   c. Does your browser allow you to create bookmarks? If so, how do you create a bookmark? How can you delete one?

   Browser dependent.

8. What is the fully abbreviated form of the IPv6 address 2620:0100:e000:0000:0000:0000:0000:8001?

   2620:100:e000::8001

10. Suppose the link between routers 1 and 2 is down in the Internet shown in Figure 8-1 on page 289. What happens if someone at site C sends a message to a user on a workstation attached to the Ethernet cable at site A? What happens if the router at site A is down? What does this tell you about designing network configurations?

    Instead of traffic going from site C to router 1 to router 2 and then to site A, traffic goes from site C to router 1 to router 3 to router 2 and then to site A.

    Network configurations are flexible and adaptive if redundancy has been designed in from the start.

12. Suppose you have 300 hosts and want to have no more than 50 hosts per subnet. What size of address block should you request from your ISP? How many /24 addresses would you need? How many subnets would you have left over from your allocation?

    The next largest subnet above 50 that is a power of 2 is 64 addresses. Because 300/50 is 6, 6 subnets of 64 would be about 2 /24-equivalent networks. The subnet mask is 255.255.255.192 or /26. There would be 2 subnets left over.

    d. a. On the local system, find two daemons running that are not listed in this chapter and explain what purpose they serve.

       System dependent.

    b. Review which services/daemons are automatically started on your system and consider which you might turn off. Are there any services/daemons in the list in Table 8-3 on page 314 you would consider adding?

       System dependent.

14. Use ssh to connect to a remote system on the local LAN using the remote system's autoconfigured link-local IPv6 address. (*Hint:* To specify the network interface to use for link-local addresses, append **%IFNAME** to the end of the address, where **IFNAME** is the local operating system's name for the interface.

    Give the command **ifconfig** or **ip addr** on the remote system to determine its IPv6 link-local address. Then use ssh to connect to that address, using the hint. An **Invalid argument** error message means you did not specify the interface. Following is an example:

    ```
    $ ssh username@fe80::1234:5678:dead:beef%eth0
    ```

9

Answers to Even-numbered Exercises

2. What are two ways you can execute a shell script when you do not have execute permission for the file containing the script? Can you execute a shell script if you do not have read permission for the file containing the script?

   You can give the name of the file containing the script as an argument to the shell (for example, **bash** *scriptfile*, where *scriptfile* is the name of the file containing the script).

   Under bash you can give either of the following commands:

   ```
   $ . scriptfile
   $ source scriptfile
   ```

   Because the shell must read the commands from the file containing a shell script before it can execute the commands, you must have read permission for the file to execute a shell script.

4. Assume you have made the following assignment:

   ```
   $ person=zach
   ```

   Give the output of each of the following commands.

   a. **echo $person**

      zach

   b. **echo '$person'**

      $person

   c. **echo "$person"**

      zach

6. Assume the **/home/zach/grants/biblios** and **/home/zach/biblios** directories exist. Specify Zach's working directory after he executes each sequence of commands. Explain what happens in each case.

   a.
   ```
   $ pwd
   /home/zach/grants
   $ CDPATH=$(pwd)
   $ cd
   $ cd biblios
   ```

   After executing the preceding commands, Zach's working directory is **/home/zach/grants/biblios**. When **CDPATH** is set and the working directory is not specified in **CDPATH**, cd searches the working directory only after it searches the directories specified by **CDPATH**.

   b.
   ```
   $ pwd
   /home/zach/grants
   $ CDPATH=$(pwd)
   $ cd $HOME/biblios
   ```

   After executing the preceding commands, Zach's working directory is **/home/zach/biblios**. When you give cd an absolute pathname as an argument, cd does not use **CDPATH**.

8. Enter the following command:

   ```
   $ sleep 30 | cat /etc/services
   ```

   Is there any output from sleep? Where does cat get its input from? What has to happen before the shell will display a prompt?

   There is no output from sleep (try giving the command **sleep 30** by itself). The **/etc/services** file provides input for cat (when cat has an argument, it does not check standard input). The sleep command has to run to completion before the shell will display a prompt.

10. Write a shell script that outputs the name of the shell executing it.

    There are many ways to solve this problem. The following solutions are all basically the same. These scripts take advantage of the **PPID** shell variable, which holds the PID number of the shell that is the parent of the process using the variable. They also use the fact that echo changes multiple sequential SPACEs to a single SPACE. The cut utility interprets multiple sequential SPACEs as multiple delimiters, so the script does not work properly without echo.

    ```
    $ cat a
    pid=$PPID
    line=$(ps | grep $pid)
    echo $line | cut --delimiter=" " --fields=4
    ```

```
$ cat a2
pid=$PPID
echo $(ps | grep $pid) | cut --delimiter=" " --fields=4

$ cat a3
echo $(ps | grep $PPID) | cut --delimiter=" " --fields=4
```

The easy solution is to give the following command:

```
$ echo $0
```

The **$0** is the first command-line token, which is usually the name of the script or program that is running (page 1133). In some cases, such as when you call the script with a relative or absolute pathname, this outcome might not be exactly what you want.

12. Add the exit status of the previous command to your prompt so it behaves similarly to the following:

```
$ [0] ls xxx
ls: xxx: No such file or directory
$ [1]
```

The following command sets up the prompt described in the question:

```
PS1='[$?] '
```

14. Implement the basename utility, which writes the last component of its pathname argument to standard output, as a bash function. For example, given the pathname **a/b/c/d**, basename writes **d** to standard output:

```
$ basename a/b/c/d
d
```

The following function is named **bn** to distinguish it from the basename utility. It behaves the same way as basename.

```
$ function bn () {
> if [ $# = 0 ]; then
>         exit 1
>     elif [ "$1" = "/" ]
>         then
>         echo /
>     else
>         echo $1 | sed 's:.*/::'
> fi
> }
```

10

Answers to Even-numbered Exercises

2. How would you communicate each of the following messages?

   a. The system is coming down tomorrow at 6:00 in the evening for periodic maintenance.

      Use the **/etc/motd** file and/or email.

   b. The system is coming down in five minutes.

      Use wall.

   c. Zach's jobs are slowing the system down drastically, and he should postpone them.

      Use write or talk.

   d. Zach's wife just had a baby girl.

      Use the **motd** file and/or email.

4. How do you kill process 1648? How do you kill all processes running kmail? In which instances do you need to work with **root** privileges?

   Give the command **kill –SIGTERM 1648** to kill process 1648. If that does not work, use SIGKILL in place of SIGTERM for a sure kill. Give the command **killall kmail** to send a SIGTERM signal to all processes running kmail. To kill any processes other than ones you own, you must run these commands with **root** privileges.

6. Give the command:

```
$ /usr/sbin/fuser -uv /
```

What does the output list? Why is it so long? Give the same command while working with **root** privileges (or ask the system administrator to do so and email you the results). How does this list differ from the first? Why is it different?

This command displays a list of processes using the root filesystem. The list is long because all files on the system are children of root; therefore this command lists all processes using any file or filesystem.

The first list shows only processes owned by the user who gives the command. When the command is run by a user working with **root** privileges, the output shows all processes. The lists are different because the system does not permit a nonprivileged user to display information about other users.

8. Take a look at **/usr/bin/lesspipe.sh**. Explain its purpose and describe six ways it works.

The **lesspipe.sh** script is a preprocessor for less. Search for **LESSOPEN** in the less man page to obtain more information on less preprocessors and postprocessors. The **lesspipe** preprocessor allows you to view archived directories and compressed files on the fly without creating intermediate files. For example, once you have set the **LESSOPEN** variable, you can view a compressed file with the command **less memo.gz** or an archived directory with the command **less myold.tar**. The **lesspipe.sh** script works with tar, tar and gzip, tar and bzip2, gzip, bzip2, zip, and cpio files. It also displays the change log when you ask less to display an **rpm** file.

10. When a user logs in, you would like the system to first check the local **/etc/passwd** file for a username and then check NIS. How do you implement this strategy?

The **/etc/nsswitch.conf** file controls the order in which sources are consulted to fulfill a request from the system. The following entry in this file causes the system to check **/etc/passwd** first and NIS second:

```
passwd:      files nis
```

11

Answers to Even-numbered Exercises

2. What does the **/etc/resolv.conf** file do? What do the **nameserver** lines in this file do?

   The **/etc/resolv.conf** file is the resolver configuration file. It provides access to DNS for Internet address resolution. The **nameserver** lines indicate which systems the local system should query to resolve hostnames into IP addresses, and vice versa.

4. What does the **..** entry in a directory point to? What does this entry point to in the root (/) directory?

   The **..** entry is a link to the parent directory. In the case of the root directory, there is no parent, so the **..** entry is a link to the root directory itself.

6. What is a FIFO? What does FIFO stand for? What is another name for a FIFO? How does a FIFO work?

   A FIFO is a special file, also called a named pipe. You read from and write to the file to read from and write to the pipe. The term FIFO stands for *first in, first out.* The first information you put in one end is the first information that comes out the other end.

8. Without using rm, how can you delete a file? (*Hint:* How do you rename a file?)

   ```
   $ mv file /dev/null
   ```

10. Why should **/var** reside on a separate partition from **/usr**?

    Files in **/var** change often, unlike files in **/usr**. When a system crashes, it is more likely that a filesystem with recently modified files will become corrupt than a stable filesystem. To lessen the chance of the data in **/usr** becoming corrupted when a system crashes, keep **/usr** on a separate partition.

12. How would you mount an ISO image so you could copy files from it without burning it to a CD?

    ```
    $ mount -t -o loop image.iso /mnt/image
    ```

    See "THE LOOP DEVICE" on the mount man page for more information.

12

Answers to Even-numbered Exercises

2. Which command would you give to update all installed packages using yum?

   ```
   # yum update
   ```

4. Suggest two advantages that rpm files have over source distributions.

   Some advantages are automatic dependency resolution, fast installation, simple uninstall, ability to query the package database, and easier deployment. Also, many software packages with complicated configuration scripts examine the local system to determine which options to compile with, so you may end up with different executables on different systems. Using distribution binary packages keeps documentation consistent with installed programs and makes it possible for developers to reproduce bugs more easily.

6. What are some steps you should take before performing an upgrade on a mission-critical server? When should you use **rpm –i** instead of **rpm –U**?

   Perform the upgrade on an identically configured spare system and see what breaks and how to fix it. In particular, look for **.rpmsave** files and see which configuration information needs to be changed manually. Before you begin the upgrade, make a complete backup copy of the server.

   Use the **–i** option when you install a new kernel. The **–i** option installs a kernel that does not overwrite the old one as **–U** would. Having the old kernel gives you a backup in case the new kernel fails.

13

Answers to Even-numbered Exercises

2. Which command would you give to cancel all print jobs on the system?

   When you are working with **root** privileges, either **lprm –** or **cancel –a** will remove all jobs from the print queues.

4. What is the purpose of sharing a Linux printer using Samba?

   Sharing a Linux printer using Samba allows Windows and OS/2 clients to send print jobs to the printer.

6. Which command lists the installed printer drivers available to CUPS?

   ```
   $ lpinfo -m
   ```

8. Assume you have a USB printer with a manufacturer-supplied PostScript printer definition file named **newprinter.ppd**. Which command would you use to add this printer to the system on the first USB port with the name USBPrinter?

10. Define a set of access control rules for a <Location> container inside **/etc/cups/cupsd.conf** that would allow anyone to print to all printers as long as they were either on the local system or in the **mydomain.com** domain.

    ```
    <Location /printers>
    Order Allow,Deny
    Allow from *.mydomain.com
    Allow from localhost
    </Location>
    ```

14

Answers to Even-numbered Exercises

2. How would you display a list of all loaded modules in the current kernel?

   Give the command **lsmod**.

4. How would you display information from the kernel about the hard disk on the first SATA channel?

   The dmesg utility displays the kernel-ring buffer, where the kernel stores messages. The following command displays the lines from this buffer that contain the string **sda**, which refers to the first SATA disk:

   ```
   $ dmesg | grep sda
   ```

6. What is a boot loader?

   A boot loader is a very small program that the bootstrap process uses as it brings a computer up from an off or reset state to a fully functional state. The boot loader frequently resides on the starting sectors of a hard disk called the MBR (master boot record).

8. You have just installed an Adaptec SCSI card. How can you find out whether it has been recognized and which entry in **/dev** represents it?

   ```
   $ dmesg | grep -i adaptec
   ```

10. How would you obtain a list of all network-related kernel parameters?

    The sysctl utility displays and configures kernel parameters at runtime. Enter the following command to display network-related kernel parameters:

    ```
    $ /sbin/sysctl -a | grep net
    ```

15

Answers to Even-numbered Exercises

2. How would you use kill to log Max off the system?

   The **–1** (one) in both of the following commands tells kill to send a TERM signal to all processes that are owned by Max:

   ```
   # su max -c 'kill -TERM -1'
   ```

   ```
   $ sudo -u max kill -TERM -1
   ```

4. Which problem does logrotate solve? Which is the primary configuration file for logrotate?

   The logrotate utility solves the problem of log files growing too large. It manages system log (and other) files automatically by rotating, compressing, mailing, and removing each file as you specify. The logrotate utility is controlled by the **/etc/logrotate.conf** file, which sets default values and can optionally specify files to be rotated.

6. If the system is less responsive than normal, what is a good first step in figuring out where the problem is?

   Run top to see if a process is using close to 100 percent of the CPU. If there is one, contact its owner or just kill the process. The user can restart the process with nice if necessary.

8. Working with **root** privileges, you are planning to delete some files but want to make sure that the wildcard expression you use is correct. Suggest two ways you could make sure you delete the correct files.

   a. Give the rm command with the **–i** flag and confirm each deletion.

b. Before giving the command, replace **rm** with **echo** on the command line. The shell expands the wildcards, and you can see which files will be deleted.

c. Redirect the output of ls with the wildcard expression to put the names of the files you want to delete in a file (named, for example, **deleteme**). When you have verified the filenames listed in the file are correct, enter the following command:

```
# rm $(cat deleteme)
```

See page 450 for information on command substitution.

16

Answers to Even-numbered Exercises

2. Which server would you set up to allow users to log in with the same username and password on all computers on a LAN?

   LDAP can provide a uniform login, regardless of which system a user logs in on. You can also use winbind (Samba) for this purpose.

4. What is a WAP, and what does it do?

   A WAP is a wireless access point; it connects a wireless network to a wired network.

6. What does a wireless bridge do?

   A wireless bridge forwards packets between wired and wireless interfaces, eliminating the need for wireless drivers.

8. What is the private address space? When would you use a private address?

   The private address space is a set of IP addresses defined by IANA and reserved for private use. Use private addresses on a LAN for systems that do not connect directly to the Internet.

10. Which file stores information about which DNS servers the system uses?

    The **/etc/resolv.conf** file stores information about which DNS servers the system uses.

12. What is Cacti? How does it communicate the information it provides?

    Cacti is a network-monitoring tool that graphs system and network information over time. It provides a Web interface for browsing the graphs it produces.

17

Answers to Even-numbered Exercises

2. What is KVM?

   KVM (Kernel-based Virtual Machine) is an open-source hypervisor that runs as part of the Linux kernel. It allows a program running in userspace (e.g., QEMU) to take advantage of hardware virtualization features of processors.

4. What is **libvirt**?

   The **libvirt** library and management tool provides a consistent interface to create, monitor, and control VMs.

6. What is the purpose of VMware Tools? Do you need to install VMware Tools when running vmplayer on a Linux system? Why or why not?

   VMware Tools is software that resides on the guest system and improves its performance, functionality, and administration of a VM.

   The Linux **open-vm-tools** package holds an open-source implementation of VMware Tools. Because VMware has merged the functionality of VMware Tools into the kernel and **open-vm-tools**, you do not need to install VMware Tools on a Linux system when **open-vm-tools** is installed.

8. What is the virt-manager utility?

   The virt-manager utility is a graphical user interface that you can use to manage virtual machines. It is based on **libvirt** and typically manages QEMU/KVM VMs, although it can manage many different types of VMs.

18

Answers to Even-numbered Exercises

2. How can you use ssh to find out who is logged in on a remote system?

   Assuming you have the same username on both systems, the following command might prompt you for your password on the remote system; it displays the output of who run on *host*:

   ```
   $ ssh host who
   ```

4. How would you use ssh to run xterm on **plum** and show the display on the local system?

   Assuming you have the same username on both systems and an X11 server running locally, the following command runs xterm on **plum** and presents the display on the local system:

   ```
   $ ssh plum xterm
   ```

   You need to use the **–Y** option if trusted X11 forwarding is not enabled.

6. When you try to connect to a remote system using an OpenSSH client and you see a message warning you that the remote host identification has changed, what has happened? What should you do?

   This message indicates that the fingerprint of the remote system is not the same as the local system remembers it. Check with the remote system's administrator to find out if something changed. If everything is in order, remove the remote system's key from the file specified in the error message and try logging in on the remote system again using ssh. You can use ssh-keygen with the **–R** option followed by the name of the remote system to remove both hashed and non-hashed entries. The system will display the first-time authentication message (page 773) again as OpenSSH verifies that you are connecting to the correct system.

8. Which single command could you give to log in as **root** on the remote system named **plum**, if **plum** has remote **root** logins disabled?

   Assuming you have the same username on both systems, the following command logs in on **plum** as **root**:

   ```
   $ ssh -t plum su -
   ```

   When you run this command, you must supply two passwords (assuming you are running the command as a user without **root** privileges and you have not set up an automatic login for ssh): yours and **root**'s. The su utility requires that its input come from standard input; the **–t** option allocates a pseudo-tty (terminal) to run su.

10. How would you use rsync with OpenSSH authentication to copy the **memos12** file from the working directory on the local system to your home directory on **plum**? How would you copy the **memos** directory from the working directory on the local system to your home directory on **plum** and cause rsync to display each file as it copied the file?

    ```
    $ rsync memos12 plum:

    $ rsync -av memos plum:
    ```

19

Answers to Even-numbered Exercises

2. What happens if you transfer an executable program file in ASCII mode?

   The file will be corrupted: Any bytes that match a NEWLINE will be altered, resulting in a program that will not execute properly.

4. How would you prevent a local user from logging in on a **vsftpd** server using her system username and password?

   Put the following line in **/etc/vsftpd/vsftpd.conf**:

   ```
   local_enable=NO
   ```

6. What is the difference between cd and lcd in ftp?

   A cd command changes the remote working directory; an lcd command changes the local working directory.

8. Why is it advantageous to run **vsftpd** in a chroot jail?

   Any program that listens for Internet connections is vulnerable to compromise. If a daemon that runs with **root** privileges is compromised, the entire system is compromised. Although the **vsftpd** daemon does not run with **root** privileges, a malicious user might still be able to use a local **root** exploit to gain **root** access. Running **vsftpd** in the restricted environment of a chroot jail makes it significantly less likely that a malicious user can compromise the system. Without **root** access, the malicious user can see only other files in the chroot jail, rendering an attack harmless.

20

Answers to Even-numbered Exercises

2. How would Max store a copy of his email in **~/mbox** and send a copy to **max@example.com**?

   Max needs to create a **~/.forward** file with the following lines:

   ```
   $ cat ~/.forward
   ~/mbox
   max@example.com
   \max
   ```

   Add the following line to **/etc/mail/sendmail.mc**:

   ```
   define(‘SMART_HOST',‘192.168.1.1')
   ```

4. What does **dnl** stand for in the m4 macro language? What are **dnl** commands used for?

   The **dnl** command, which stands for **delete to new line**, instructs the compiler to ignore anything on a line following the **dnl**. It is used to set off comments.

5. SpamAssassin is installed on your mail server, with the threshold set to an unusually low value of 3, resulting in a lot of false positives. What rule could you give to your mail client to allow it to identify spam with a score of 5 or higher?

   Select messages where the header **X-Spam-Level** contains *****.

7. Your company's current mail server runs on a commercial UNIX server, and you are planning to migrate it to Linux. After copying the configuration files across to the Linux system, you find that it does not work. What might you have forgotten to change?

You must change the OSTYPE directive in the **sendmail.mc** file to OSTYPE(‘linux'). Some paths in the configuration file might also need to be changed.

8. Assume a script stores certain information in a variable named **RESULT**. What line could you put in the script that would send the contents of **RESULT** to the email address specified by the first argument on the command line?

21

Answers to Even-numbered Exercises

2. How would you prevent NIS from exporting the **root** user and other system users to clients?

   Ensure that these users have UIDs and GIDs lower than the values **MINUID** and **MINGID** are set to in **/var/yp/Makefile**.

4. Why does the **/etc/passwd** file need two NIS maps?

   Password information is looked up two ways: by UID and by username. Because dbm files, which implement NIS maps, have a single index, you need two dbm files (maps) to enable the two kinds of lookups on NIS user information.

6. What is the basic unit of information in an LDAP directory? What is the structure of an attribute?

   An entry is the basic unit of information in an LDAP directory.

   Each attribute has a name (an attribute type or description) and one or more values.

8. How can you determine whether the working directory is the home directory of an NIS user?

   ```
   $ ypcat passwd | grep $(pwd)
   ```

10. Where is the LDAP **device** object class defined? Which of its attributes are mandatory and which are optional?

    The **device** object class is defined in the **/etc/openldap/schema/core.ldif** file. Its mandatory attribute is **cn**. Its optional attributes are **serialNumber**, **SeeAlso**, **owner**, **ou**, **o**, **l**, and **description**.

22

Answers to Even-numbered Exercises

2. Which command would you give to mount on the local system the **/home** directory hierarchy that resides on the file server named **plum**? Assume the mounted directory hierarchy will appear as **/plum.home** on the local system. How would you mount the same directory hierarchy if it resided on the fileserver at 192.168.1.1? How would you unmount **/home**?

```
$ sudo mount plum:/home /plum.home
$ sudo mount 192.168.1.1:/home /plum.home
$ sudo umount /home
```

4. Which command line lists the currently mounted NFS directory hierarchies?

```
$ mount | grep nfs4
```

or

```
$ df -t nfs4
```

6. From a server, how would you allow readonly access to **/opt** for any system in **example.com**?

   Place the following line in **/etc/exports**:

8. Describe the difference between the **root_squash** and **all_squash** options in **/etc/exports**.

   The **root_squash** option maps **root** to **nfsnobody**; **all_squash** maps all users to **nfsnobody**.

10. Some diskless workstations use NFS as swap space. Why is this approach useful? What is the downside?

    Because it has no disk space, a diskless workstation has no swap space. The only choice is to use NFS; if it did not use NFS for swap space, the workstation would be limited by the amount of its physical memory (RAM).

    Swapping in general is slow because disks are much slower than RAM. NFS is even slower than a local disk; any process that uses an NFS swap space will spend a long time waiting for pages to be swapped in.

12. What does the mount **nosuid** option do? Why would you want to use this option?

    The **nosuid** option forces setuid executables in the mounted directory hierarchy to run with regular permissions on the local system.

    Giving a user the ability to run a setuid program can give that user the ability to run a program with **root** privileges. Normally you do not want an ordinary user running a program as a privileged user.

23

Answers to Even-numbered Exercises

2. What steps are required for mapping a Windows user to a Linux user?

   Set the **username map** parameter in **smb.conf** to point to the map file, frequently **/etc/samba/smbusers**, add the user to **/etc/samba/smbusers**, and use smbpasswd with the **–a** option to create and assign a Samba password to the user.

4. What is the purpose of the **[homes]** share? Should this share be browseable? Why?

   The **[homes]** share implicitly shares the home directory of each user without having to define specific shares. It should not be browseable because being browseable exposes usernames (names of the directories in **[homes]**), making it easier for a malicious user to break into the system.

6. Which configuration changes would you need to apply to routers if you wanted to allow SMB/CIFS browsing across multiple subnets without configuring master browsers?

   Routers do not usually allow broadcast packets to propagate between subnets. Browsing, which uses broadcasts, is blocked by a router. Configuring the router to forward broadcast packets between subnets would allow browsing across subnets.

24

Answers to Even-numbered Exercises

2. What kind of DNS record is likely to be returned when a Web browser tries to resolve the domain part of a URI?

   An A (address) record points to a domain.

4. How would you find the IP address of example.com from the command line?

   ```
   $ host example.com
   ```

   or

   ```
   $ dig example.com
   ```

   ```
   DNS1=192.168.1.254
   DNS2=1.2.3.4
   ```

6. How would you instruct a DNS server to respond only to queries from the **137.44.*** IP range?

   Add the following line to the Options clause in **/etc/named.conf**:

   ```
   allow-query { 137.44.0.0/24; };
   ```

8. How would you set up a private domain name hierarchy that does not include any of the official InterNIC-assigned domain names?

   Set up a DNS cache that defines the zone **.** (period) clause explicitly, rather than relying on the hint file.

10. It is often irritating to have to wait for DNS records to update around the world when you change DNS entries. You could prevent this delay by setting the TTL to a small number. Why is setting the TTL to a small number a bad idea?

Setting the TTL to a small number prevents DNS caches from holding DNS entries for very long. Small TTL values place a large load on the local DNS server because every query about the domain is forwarded to the local server.

25

Answers to Even-numbered Exercises

2. How can you tell whether **firewalld** is running?

```
$ systemctl status firewalld.service
```

or

```
$ firewall-cmd --state
```

4. Which firewall-cmd command would verify that the FTP service was trusted in the runtime configuration? In the permanent configuration?

```
$ su -c 'firewall-cmd --list-services'
```

```
$ su -c 'firewall-cmd --permanent --list-services'
```

6. Define an iptables rule that will reject incoming connections on the TELNET port.

```
$ su -c 'iptables --append FORWARD --sport telnet --jump REJECT'
```

8. Write an iptables (IPv4) rule that silently blocks incoming SMTP connections from **10.10.0.10**.

```
$ su -c 'iptables --append FORWARD -p tcp --dport smtp --source 10.10.0.10 --jump DROP'
```

26

Answers to Even-numbered Exercises

2. What is the function of the document root? Where is it located by default? How would you change the location of the document root?

   The root of the directory hierarchy that Apache serves content from is called the *document root*. By default it is located at **/var/www/html**. This directory on the server maps to **/** so it appears to users who are browsing a Web site as the root directory.

4. How would you instruct an Apache server to listen on port 81 instead of port 80?

   In **httpd.conf**, change the directive

   ```
   Listen 80
   ```

   to

   ```
   Listen 81
   ```

   Place the following directives in **httpd.conf**:

   ```
   UserDir website
   UserDir disabled
   UserDir enabled sam
   ```

6. Apache must be started with **root** privileges. Why? Why does this action not present a security risk?

   By default, **httpd** listens on port 80, which is a privileged port. Only a process with **root** privileges can use privileged ports, so you must start Apache with **root** privileges. Starting Apache with **root** privileges does not pose a security risk because Apache uses child processes running as **apache** to serve pages. The original **httpd** process running with **root** privileges remains but does not interact over the network.

8. What does the ServerName directive do? Which value can you use as a ServerName if you want to experiment with an Apache server locally (on the server system)?

   The ServerName directive establishes a name for the server.

   If you do not need to access an Apache server from other systems, you can specify a ServerName of 127.0.0.1, the address of **localhost**.

10. Why is it more efficient to run scripts using **mod_perl** than to run them through CGI?

    Running a CGI script requires system calls to **fork**() and **exec**() to create a new process. Once the process has finished (which, in the case of CGI scripts, is usually very shortly after it has started), it terminates. A script run from a module does not have this overhead because it runs inside the Apache server process.

12. Some Web sites generate content by retrieving data from a database and inserting it into a template using PHP or CGI each time the site is accessed. Why is this practice often a poor idea?

    In many cases, the same data is generated each time a given page is accessed, unnecessarily consuming CPU and disk resources for each access. Using resources unnecessarily can result in pages being unavailable when the system load is high.

14. Part of a Web site is a private intranet. Describe how you would prevent people outside the company's internal 192.168.0.0/16 network from accessing this site. The site is defined as follows:

```
<VirtualHost *>
  ServerName example.com
  DocumentRoot /var/www
  <Directory /var/www/intranet>
      AllowOverride AuthConfig
  </Directory>
</VirtualHost>
```

    a. Add the following to the <Directory /var/www/intranet> container:

```
Order deny,allow
Deny from all
Allow from 192.168.
```

    b. Create **/var/www/intranet/.htaccess** with the following:

```
Order deny,allow
Deny from all
Allow from 192.168.
```

16. What does CGI do? What are the characteristics of a CGI program?

    The CGI (Common Gateway Interface) allows external application programs to interface with Web servers.

    Any program can be a CGI program if it runs in real time and relays its output to the requesting client.

27

Answers to Even-numbered Exercises

2. The special parameter "**$@**" is referenced twice in the **out** script (page 989). Explain what would be different if the parameter "**$***" were used in its place.

   If you replace "**$@**" with "**$***" in the **out** script, cat or less would be given a single argument: a list of all files you specified on the command line enclosed within single quotation marks. This list works when you specify a single filename. When you specify more than one file, the shell reports **No such file or directory** because there is not a file whose name matches the string you specified on the command line (SPACEs are not considered special characters when they are enclosed within single quotation marks).

4. Write a function that takes a single filename as an argument and adds execute permission to the file for the user.

```
$ function perms () {
> chmod u+x $1
> }
```

   a. When might such a function be useful?

      When you are writing many shell scripts, it can be tedious to give many chmod commands. This function speeds up the process.

   b. Revise the script so it takes one or more filenames as arguments and adds execute permission for the user for each file argument.

```
$ function perms () {
> chmod u+x $*
> }
```

c. What can you do to make the function available every time you log in?

Put the function in **~/.bash_profile** and/or **~/.bashrc** to make it available each time you log in (using bash).

d. Suppose that, in addition to having the function available on subsequent login sessions, you want to make the function available in your current shell. How would you do so?

Use source to execute the file you put the function in. For example:

```
$ source ~/.bash_profile
```

6. Write a shell script that displays the names of all directory files, but no other types of files, in the working directory.

There are many ways to solve this problem. The **listdirs** script uses file to identify directory files and grep to pull them out of the list. Then sed removes everything from file's output, starting with the colon.

```
$ cat listdirs
file "$@"      |
grep directory |
sed 's/:.*//'
```

8. Enter the following script named **savefiles**, and give yourself execute permission to the file:

```
$ cat savefiles
#! /bin/bash
echo "Saving files in working directory to the file savethem."
exec > savethem
for i in *
        do
        echo "================================================="
        echo "File: $i"
        echo "================================================="
        cat "$i"
        done
```

a. Which error message do you receive when you execute this script? Rewrite the script so that the error does not occur, making sure the output still goes to **savethem**.

You receive the following error message:

```
cat: savethem: input file is output file
```

To eliminate the error message, add the following lines after the line with **do** on it:

```
if [ $i == "savethem" ] ; then
        continue
fi
```

b. What might be a problem with running this script twice in the same directory? Discuss a solution to this problem.

Each time you run **savefiles**, it overwrites the **savethem** file with the current contents of the working directory. When you remove a file and run **savefiles** again, that file will no longer be in **savethem**. If you want to keep an archive of files in the working directory, you need to save the files to a new file each time you run **savefiles**. If you prefix the filename **savethem** with **$$**, you will have a unique filename each time you run **savefiles**.

10. Using the find utility, perform the following tasks:

a. List all files in the working directory and all subdirectories that have been modified within the last day.

```
$ find . -mtime -1
```

b. List all files you have read access to on the system that are larger than 1 megabyte.

```
$ find / -size +1024k
```

c. Remove all files named **core** from the directory structure rooted at your home directory.

```
$ find ~ -name core -exec rm {} \;
```

d. List the inode numbers of all files in the working directory whose filenames end in **.c**.

```
$ find . -name "*.c" -ls
```

e. List all files you have read access to on the root filesystem that have been modified in the last 30 days.

```
$ find / -xdev -mtime -30
```

12. Write a script that takes the name of a directory as an argument and searches the file hierarchy rooted at that directory for zero-length files. Write the names of all zero-length files to standard output. If there is no option on the command line, have the script delete the file after displaying its name, asking the user for confirmation, and receiving positive confirmation. A **–f** (force) option on the command line indicates that the script should display the filename but not ask for confirmation before deleting the file.

The following script segment deletes only ordinary files, not directories. As always, you must specify a shell and check the arguments.

```
$ cat zerdel
if [ "$1" == "-f" ]
        then
                find $2 -empty -print -exec rm -f {} \;
        else
                find $1 -empty -ok rm -f {} \;
fi
```

14. Generalize the script written in exercise 13 so the character separating the list items is given as an argument to the function. If this argument is absent, the separator should default to a colon.

    This script segment takes an option in the form **–d***x* to specify the delimiter *x:*

```
$ cat nodel
if [[ $1 == -d? ]]
        then
                del=$(echo $1 | cut -b3)
                shift
        else
                del=:
fi
IFS=$del
set $*
for i
        do
        echo $i
done
```

16. Rewrite **bundle** (page 1015) so the script it creates takes an optional list of filenames as arguments. If one or more filenames are given on the command line, only those files should be re-created; otherwise, all files in the shell archive should be re-created. For example, suppose all files with the filename extension **.c** are bundled into an archive named **srcshell**, and you want to unbundle just the files **test1.c** and **test2.c**. The following command will unbundle just these two files:

```
$ bash srcshell test1.c test2.c
```

```
$ cat bundle2
#!/bin/bash
# bundle: group files into distribution package

echo "# To unbundle, bash this file"
for i
do
        echo 'if echo $* | grep -q' $i '|| [ $# = 0 ]'
                echo then
                echo "echo $i 1>&2"
                echo "cat >$i <<'End of $i'"
                cat $i
                echo "End of $i"
        echo fi
done
```

18. In principle, recursion is never necessary. It can always be replaced by an iterative construct, such as **while** or **until**. Rewrite **makepath** (page 1066) as a nonrecursive function. Which version do you prefer? Why?

```
function makepath2()
{
wd=$(pwd)
pathname=$1

while [[ $pathname = */* && ${#pathname} > 0 ]]
        do
        if [[ ! -d "${pathname%%/*}" ]]
                then
                mkdir "${pathname%%/*}"
        fi
        cd "${pathname%%/*}"
        pathname="${pathname#*/}"
        done
if [[ ! -d $pathname && ${#pathname} > 0 ]]
        then
        mkdir $pathname
fi
cd $wd
}
```

The recursive version is simpler: There is no need to keep track of the working directory and you do not have to handle the task of making the final directory separately.

20. Write a function that takes a directory name as an argument and writes to standard output the maximum of the lengths of all filenames in that directory. If the function's argument is not a directory name, write an error message to standard output and exit with nonzero status.

```
$ function maxfn () {
> declare -i max thisone
> if [ ! -d "$1" -o $# = 0 ]
>     then
>     echo "Usage: maxfn dirname"
>     return 1
> fi
>
> max=0
> for fn in $(/bin/ls $1)
>     do
>     thisone=${#fn}
>     if [ $thisone -gt $max ]
>         then
>         max=$thisone
>     fi
>     done
> echo "Longest filename is $max characters."
> }
```

22. Write a function that lists the number of ordinary files, directories, block special files, character special files, FIFOs, and symbolic links in the working directory. Do this in two different ways:

    a. Use the first letter of the output of **ls –l** to determine a file's type.

```
$ function ft () {
 > declare -i ord=0 dir=0 blk=0 char=0 fifo=0 symlnk=0 other=0
 >
 > for fn in *
 >     do
 >     case $(ls -ld "$fn" | cut -b1) in
 >         d)
 >             ((dir=$dir+1))
 >             ;;
 >         b)
 >             ((blk=$blk+1))
 >             ;;
 >         c)
 >             ((char=$char+1))
 >             ;;
 >         p)
 >             ((fifo=$fifo+1))
 >             ;;
 >         l)
 >             ((symlnk=$symlnk+1))
 >             ;;
 >         a-z)
 >             ((other=other+1))
 >             ;;
 >         *)
 >             ((ord=ord+1))
 >             ;;
 >     esac
 >     done
 >
 > echo $ord ordinary
 > echo $dir directory
 > echo $blk block
 > echo $char character
 > echo $fifo FIFO
 > echo $symlnk symbolic link
 > echo $other other
 > }
```

b. Use the file type condition tests of the *[[expression]]* syntax to determine a file's type.

```
$ function ft2 () {
 > declare -i ord=0 dir=0 blk=0 char=0 fifo=0 symlnk=0 other=0
 >
 > for fn in *
 >     do
 >         if [[ -h $fn ]]
 >             then ((symlnk=$symlnk+1))
 >         elif [[ -f $fn ]]
 >             then ((ord=ord+1))
 >         elif [[ -d $fn ]]
 >             then ((dir=$dir+1))
 >         elif [[ -b $fn ]]
 >             then ((blk=$blk+1))
 >         elif [[ -c $fn ]]
 >             then ((char=$char+1))
 >         elif [[ -p $fn ]]
 >             then ((fifo=$fifo+1))
 >         else
 >             ((other=other+1))
 > fi
 >     done
 >
 > echo $ord ordinary
 > echo $dir directory
 > echo $blk block
 > echo $char character
 > echo $fifo FIFO
 > echo $symlnk symbolic link
 > echo $other other
 > }
```

28

Answers to Even-numbered Exercises

2. Write and run a Python program that you store in a file. The program should demonstrate how to prompt the user for input and display the string the user entered.

```
$ cat mypy.py
#!/usr/bin/python

inp = raw_input('Enter the name of a month: ')
print 'You entered ' + inp

$ python mypy.py
Enter the name of a month: June
You entered June
```

or

```
$ chmod 755 mypy.py
$ ./mypy.py
```

4. Using the Python interactive shell, use a **for** control structure to iterate through the elements of the list you instantiated in exercise 3 and display each abbreviated name followed by a period on a line by itself. (*Hint:* The period is a string.)

```
>>> for nam in mon:
...     print nam + '.'
...
Jan.
Feb.
Mar.
Apr.
May.
Jun.
```

6. Instantiate a dictionary in which the keys are the months in the third quarter of the year and the values are the number of days in the corresponding month. Display the dictionary, the keys, and the values. Add the tenth month of the year to the dictionary and display the value of that month only.

```
>>> days = {'July': 31, 'August': 31, 'September': 30}
>>> days
{'September': 30, 'July': 31, 'August': 31}
>>> days.keys()
['September', 'July', 'August']
>>> days.values()
[30, 31, 31]
>>> days['October'] = 31
>>> days['October']
31
```

8. Write and demonstrate a Lambda function named **stg()** that appends **.txt** to its argument. What happens when you call the function with an integer?

```
>>> stg = lambda b: b + '.txt'
>>> stg('aaa')
'aaa.txt'
```

When called with an integer, the function returns an error when it tries to catenate an integer and a string:

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

10. Define a function named **cents2** that returns its argument divided by 100 exactly (and includes decimal places if necessary). Make sure your function does not truncate the answer. For example:

```
>>> cents2(12345)
123.45

>>> def cents2(val):
...     return(val / 100.)
```

12. Why does the following assignment statement generate an error?

```
>>> x.y = 5
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'x' is not defined
```

The **x.y** is not a valid name; Python parses it as "apply method **y** to object **x**." Python reports that **x** is not defined.

14. Use a list comprehension to display the numbers from 1 through 30 inclusive that are divisible by 3.

```
>>> [n for n in range(1,31) if n % 3 == 0]
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30]
```

16. Rewrite exercise 15 to call the function with a random number between 0 and 10 inclusive. (*Hint:* The **randint** function in the **random** library returns a random number between its two arguments inclusive.)

```
$ cat number_guess2.py
#!/usr/bin/python
from random import randint

def guessANumber(val):
    guess = int(input('Enter your guess: '))
    if guess > val:
        print 'Too high.'
        return 1
    elif guess < val:
        print 'Too low.'
        return -1
    else:
        print 'Got it!'
        return 0

val = randint(0,10)
while (guessANumber(val) != 0):
    print 'Guess again.',
```

18. Write a function that counts the vowels (**aeiou**) in a string the user inputs. Make sure it counts upper- and lowercase vowels. Then write a routine that calls the function and displays the following output.

```
$ ./count_vowels.py
Enter some words: Go East young man!
The string "Go East young man!" has  6 vowels in it.
```

```
$ cat count_vowels.py
#!/usr/bin/python
def countVowels(my_string):
    count = 0
    for s in my_string:
        if s.lower() in 'aeiou':
            count += 1
    return count

stg = raw_input('Enter some words: ')
print 'The string \"' + stg + '\"' + ' has ',
print countVowels(stg),
print 'vowels in it.'
```

Following is an alternative function that performs the same task:

```
$ cat count_vowels2.py
#!/usr/bin/python
def countVowels(my_string):
    my_string = my_string.lower()
    return len(my_string) - len(my_string.translate(None,'aeiou'))

stg = raw_input('Enter some words: ')
print 'The string \"' + stg + '\"' + ' has ',
print countVowels(stg),
print 'vowels in it.'
```

29

Answers to Even-numbered Exercises

2. Using MariaDB interactively, create a database named **dbsam** that the user named **sam** can modify and grant privileges on. Set up Sam's password to be **porcupine**. The MariaDB user named **root** has the password **five22four**.

```
$ mysql -u root -p
Enter password: five22four
...
MariaDB [maxdb]> CREATE DATABASE dbsam;
Query OK, 1 row affected (0.00 sec)

MariaDB [maxdb]> GRANT           ALL PRIVILEGES
    ->                   ON      dbsam.* to 'sam'
    ->                   IDENTIFIED BY 'porcupine'
    ->                   WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)
```

4. Which commands would you use to set up a table in **dbsam** (created in exercise 2) named **shoplist** with the following columns of the specified types: **day** [DATE], **store** [CHAR(20)], **lettuce** [SMALLINT], **soupkind** [CHAR(20)], **soupnum** [INTEGER], and **misc** [VARCHAR(40)]?

```
USE dbsam;
CREATE TABLE   shoplist (
               day      DATE,
               store    CHAR(20),
               lettuce  SMALLINT,
               soupkind CHAR(20),
               soupnum  INTEGER,
               misc     VARCHAR(40)
               );
```

6. List two ways you can specify the name of a specific MariaDB database to work with.

   You can specify the name of the database you are working with in your **~/.my.cnf** file or by using a USE statement.

8. Assume you are working with the **people** table in the **maxdb** database described in this chapter. Write a query that lists the names of all the people and their hire dates sorted by their names.

```
SELECT          name,
                hired
        FROM    people
        ORDER BY name;
```