12

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 5 minutes.

   Use wall.

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

   Use write.

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

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

4. How would you allow a user to execute a specific, privileged command without giving the user the **root** password or permission to use sudo to run any command with **root** privileges?

   You can create a setuid program that belongs to a group that only the user who is to execute it belongs to and that has no permissions for other users. Alternatively, you can edit the **sudoers** file to grant the user permission to use sudo to execute the command.

6. What does the **/etc/event.d/logd** file do and what starts it? What does the **respawn** keyword in this file mean?

   The **/etc/event.d/logd** file controls the Upstart **logd** service, which starts and monitors the **logd** daemon. The comments in the file indicate that **init** starts the **logd** service automatically. The **stop on runlevel** lines stop the service when the system is being brought down (runlevel 0), when it enters runlevel 1, and when it is being rebooted (runlevel 6). Output is sent to the console. The **respawn** keyword causes **init** to restart **logd** if it stops at a point other than when the system enters runlevel 0, 1, or 6.

8. Give the command

```
$ /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 run by a user 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.

10. Take a look at **/usr/bin/lesspipe**. Explain its purpose and give six ways it works.

The **lesspipe** 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 (give the command **lesspipe** and execute the first command line it displays), you can view a compressed file with the command **less memo.gz** or an archived directory with the command **less myold.tar**. The **lesspipe** script works with tar, tar and gzip, tar and bzip2, gzip, bzip2, zip, and cpio files.When you ask less to display a deb file, it displays information about the file, including a list of files it contains.

12. 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 first **/etc/passwd** and then NIS:

```
passwd:      files nis
```