

Answers to Even-Numbered Exercises 17

from page 1077

1. What option should you use with `fsck` if you want to review the status of your filesystems without making any changes to them? How does `fsck` determine what devices to check if you do not specify any on the command line?

2. How does single-user mode differ from multiuser mode?

When a system is in single-user mode, you can log in only at the console. Not all of the filesystems are mounted, and many daemons are not running. With the system in multiuser mode you can log in at any terminal or workstation that is set up for login, most or all of the filesystems are mounted, and all of the daemons that your system is set up to run are running.

3. 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.
- b. The system is coming down in 5 minutes.
- c. Jenny's jobs are slowing the system down drastically, and she should postpone them.
- d. Alex's wife just had a baby girl.

4. If Alex belongs to five groups—**inhouse**, **pubs**, **sys**, **other**, and **supers**—how would his group memberships be represented? Assume that **inhouse** is his primary group. How would Alex create a file that belongs to the group **pubs**?

The `/etc/group` file lists all groups, one per line, followed by members of each group on the same line. Alex would be listed on each line that defined a group he was a member of. He would not be listed on the line with his primary group; instead, the primary group GID number would be listed in the fourth field of his entry in `/etc/passwd`.

There are two ways Alex can create a file. First, he can create a file and use `chgrp groupname filename` to change the group association of the file. The second way is for Alex to use `newgrp groupname` to change his group association and then create the file.

5. How can you identify the user ID of another user on your system? What is the user ID of **root**?
6. How can you redirect the output of the `find` command so that whatever it sends to the standard error disappears?

Under `bash`, the following command sends standard error to `/dev/null`, which makes it disappear.

```
$ find . -name "*.c" 2> /dev/null
```

7. How many links does a file have? What happens when you add a hard link to a file? What happens when you add a symbolic (soft) link?
8. How would you add a printer named **quark** that is on a remote machine named **physics**? How would you add a printer named **greens** if it were attached to the local machine on serial port B at 19,200 bps?

Click GNOME/K menu: System Settings ⇒ Printing to run `printconf-gui`. Select New, specify a Queue Name (for example `quarkque1`), click Unix Printer, Specify a **physics** as a Server and optionally a Queue if you are not using the default queue on **quark**, and select a Printer.

As just shown, but click Local Printer, Custom Device, and specify `/dev/ttyS1`.

You can use the following command to see how a serial port is set up:

```
$ cat /proc/tty/driver/serial
```

9. What are the differences between a character device and a block device?

10. What is a named pipe? Give an example of how one is used.

A named pipe (FIFO) is used by programs to communicate with other programs asynchronously. One program writes to the pipe, and the other reads from it. You can use the following scripts to demonstrate how a named pipe works. You can use any text file in place of `/etc/services`.

```
$ cat feedmypipe
#!/bin/bash

for word in $(cat /etc/services)
do
    echo $word > /tmp/mypipe
    sleep 1
done

$ cat readmypipe
#!/bin/bash

while /bin/true
do
    cat /tmp/mypipe
done

$ cat /proc/tty/driver/serial
```

The `feedmypipe` script sends one word per second from the `/etc/services` file to the pipe named `/tmp/mypipe`. The `readmypipe` reads words from `/tmp/mypipe`. Implement the pipe as follows:

```
$ mkfifo /tmp/mypipe
$ feedmypipe &
$ readmypipe
$ readmypipe
#
/etc/services:
#
$Id:
services,v
1.31
2002/04/03
...
```

11. How would you mount the `/export/apps` filesystem from a server named `achilles` to a client named `perseus`? Give the commands for the client and the server machines.

12. Implement a local firewall on your system

Select GNOME/K menu: System Settings ⇒ Security Level to run `redhat-config-securitylevel`. Choose High, Medium, or No Firewall and click OK.

Advanced Exercises

13. A process is using 98 percent of the CPU time. How do you identify the process and determine whether you should kill it?

14. What are the differences between a FIFO and a socket?

Sockets handle interprocess communication for networking, while FIFOs allow communication between other types of processes.

15. Develop a strategy for coming up with a password that an intruder would not be likely to guess but that you will be able to remember.

16. Develop a backup strategy that is executed by cron and includes the following components:

- a. A level 0 backup is done once per month.
- b. A level 2 dump is performed one day per week.
- c. A level 5 dump is performed every day that neither a level 0 nor a level 2 dump is performed.

```
# cat /var/spool/cron/root
10 2 1 * * /sbin/dump -0
10 2 2-31 * 6 /sbin/dump -2
10 2 2-31 * 1-5,7 /sbin/dump -5
```

In the worst case, how many restores would you have to perform to recover a file that was dumped using the preceding schedule?

In the worst case, you would need to restore from three backups: the most recent level 0 dump, the most recent level 2 dump, and finally the most recent level 5 dump.

17. How would you restrict access to a tape drive on your system so that only certain users could read and write tapes?

18. Design and implement a job that runs every night at 11:30 and removes from the `/home` filesystem all files named `core` that are more than a week old.

Put the following line in `root`'s crontab file (`/var/spool/cron/root`):

```
30 23 * * * /usr/bin/find /home -name "core" -mtime +7 -exec /bin/rm {} \;
```

The one-line script on page 1072 performs a similar function but does not remove necessary system files named `core`.

19. Give the command

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

What is this a list of? Why is it so long? Give the same command as `root` (or ask your system administrator to do so and mail you the results). How does this list differ from the first? Why is it different?

20. When it puts files in a `lost+found` directory, `fsck` has lost the directory information for the files and thus has lost the names of the files. Each file is given a new name, which is the same as the inode number for the file:

```
$ ls -lg lost+found
-rw-r--r-- 1 alex pubs  110 Jun 10 10:55 51262
```

What can you do to identify these files and restore them?

Run `file` on the files in `lost+found`. View any readable file (text, C program, ASCII, shell script, and so on) with `more` or a text editor. You can run strings on an executable file to try to figure out which program it is. An `ls -l` command displays the name of the owner of the file; gather additional information from the owner regarding which files are missing.

21. How would you allow a user to execute privileged commands without giving the user the Superuser password?
22. What do the letters of the `su` command stand for? (*Hint*: It is not Superuser.) What can you do with `su` besides give yourself Superuser privileges? How would you log in as Alex if you did not know his password but knew the `root` password? How would you establish the same environment that Alex has when he first logs on?

The letters stand for substitute user. You can give yourself privileges of any user whose password you know (or any user when you are running as `root`). You can also execute a command (other than a shell) as the specified user.

To log in as Alex, you would first log in as **root**, and then give the command **su alex**, or **su - alex** to establish the same environment that Alex has when he logs in.

23. Take a look at **/usr/bin/lesspipe.sh**, and explain what it does and six ways it works.

24. Use **at** to reboot the system

- a. At 3 A.M. the following morning.

```
# at 0300
warning: commands will be executed using (in order) a) $SHELL b) ...
at> /sbin/shutdown -r now
at>CONTROL-D
```

- b. Next Friday at 1 minute past midnight.

```
# at 00:01 friday
warning: commands will be executed using (in order) a) $SHELL b) ...
at> /sbin/shutdown -r now
at>CONTROL-D
```

- c. Two weeks from tomorrow at the current time.

```
# at tomorrow + 2 weeks
warning: commands will be executed using (in order) a) $SHELL b) ...
at> /sbin/shutdown -r now
at>CONTROL-D
```

- d. In 30 minutes, using the TC Shell.

```
# at now + 30 minutes
warning: commands will be executed using (in order) a) $SHELL b) ...
at> /sbin/shutdown -r now
at>CONTROL-D
```

25. Give a command that will make a level 0 dump of the **/usr** filesystem to the first tape device on the system. What command would you use to take advantage of a drive that supports compression? What command would place a level 3 dump of the **/var** filesystem immediately after the level 0 dump on the tape?

26. How would you create a new directory **/home/shared/billken** so that users Bill and Kendra can create files in the directory? Any new files or subdirectories that either user creates should automatically be writable by either user. No one else should have access to this directory or the files within it. Modify this directory so that the group **spiffy** has readonly access to all files and directories within **billken**.

Create a group named **billken** and add Bill and Kendra to the group (in **/etc/group**). Create the **/home/shared/billken** directory with group **billken** (**chgrp**) read, write, and execute permission for the group, but no permissions for other (**chmod 770**). Before Bill or Kendra works or creates files in the new directory, he or she must give a **newgrp billken** command so that the new files belong to the **billken** group.

Without taking advantage of an ACL (access control list), you cannot add permissions for a second group. Red Hat 8 and 9 do not turn on ACLs but include the **attr** and **acl** packages in case you want to experiment with ACLs. See the man pages for **getfacl** and **setfacl** for more information.

27. Why are **setuid** shell scripts inherently unsafe?
28. A utility named **/usr/bin/netc1k** accepts a connection over the network and quits once the connection is dropped. How can you use the built-in functionality of Red Hat to make this program run so that it restarts automatically (without modifying **netc1k**)? (*Hint*: It should run only in multiuser mode.)

Put the following line in **/etc/inittab**:

```
nc:5:respawn:/usr/bin/netc1k
```

29. A process is consuming a great deal of memory. How do you determine how much physical memory it is using and what percentage this is of the total memory?
30. When a user logs in, you would like the system to check for a login name in the local **/etc/passwd** file first and then to check NIS. How do you implement this strategy?

The **/etc/nsswitch.conf** file (page 1022) controls the order in which the various sources are consulted to fulfill a request from the system. The following entry in this file causes the system to check the **passwd** file (page 951) and then NIS:

```
passwd:      files nis
```

31. Implement a local firewall on your system without using **gnome-lokkit** or **redhat-config-securitylevel**.