

Answers to Even-Numbered Exercises

3

from page 98

1. What commands can you use to determine who is logged in on a specific terminal?
2. How can you keep other users from using `write` to communicate with you? Why would you want to?

Give the command `mesg n` to keep other ordinary users from writing to your terminal. You can take this action when you do not want to be disturbed or are viewing something on your screen that you do not want overwritten.
3. What command sends the files `chapter1`, `chapter2`, and `chapter3` to the printer?
4. List some differences between `talk` and `write`. Why are three different communications utilities (`talk`, `write`, `pine`) useful? Describe a situation in which it makes sense to use
 - a. `pine` instead of `talk` or `write`
 - b. `talk` instead of `write`
 - c. `write` instead of `talk`

The `write` utility

- a. Displays a line on the other user's screen only after you press RETURN.
- b. Does not work over a network.
- c. Mixes messages from you and the person you are communicating with.

The `talk` utility

- a. Displays each character as you type it.
- b. Works over a network.
- c. Separates messages from you and the person you are communicating with.

You can always send e-mail (using `pine` or another e-mail program) regardless of whether the user receiving the email is logged in and regardless of whether the user logs in on the same machine as you do.

In order to use `talk`, the user you want to communicate with must be logged in on a machine connected to a network you are connected to and must be accepting messages (`mesg`).

In order to use `write`, the user you want to communicate with must be logged in on the same machine as you and accepting messages (`mesg`).

5. Show how to use `pine` to send a single mail message to **agnes** on the system named **cougar** and to **jim** on the system named **ucsf**. Assume that your computer has network links to **cougar** and **ucsf**.
6. What happens when you give the following commands if the file named **done** already exists?

```
$ cp to_do done
$ mv to_do done
```

Either command overwrites **done** with the contents of **to_do**.

7. How can you find out which utilities are available on your system for editing files? What utilities are there for editing on your system?
8. How can you find the phone number for Ace Electronics in a file named **phone** that contains a list of names and phone numbers? What command can you use to display the entire file in alphabetical order? How can you remove adjacent duplicate lines from the file?

```
$ grep "Ace Electronics" phone
$ sort phone
$ uniq phone
```

9. 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.
10. Create a `.plan` file in your home directory. Does `finger` on your system display the contents of your `.plan` file?
The answer is system dependent.
11. What is the result of giving the `which` utility the name of a command that resides in a directory that is *not* in your search path?
12. Are any of the utilities discussed in this chapter located in more than one directory on your system? If so, which ones?
No. However, some commands that are built into a shell have an executable counterpart that exists as a file (for example, `echo`).
13. Experiment by calling the `file` utility with names of files in `/usr/bin`. How many different types of files are there?
14. What command can you use to look at the first few lines of a file named `status.report`? What command can you use to look at the end of the file?

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

Advanced Exercises

15. Recreate the `colors.1` and `colors.2` files used in Figure 3-7 on page 68. Test your files by running `diff -u` on them, and see whether you get the same results as in the figure.
16. Use the pine mailer to create a new folder named `tmp-mail`. Then describe how to move a message from the folder `sent-mail` to the folder `tmp-mail`.
Create a new folder with these commands: L (FOLDER LIST), > (View Collection), A (Add), `tmp-mail` RETURN.
Move a message from `sent-mail` to `tmp-mail` with these commands: L, >, N/P until `sent-mail` is highlighted, > (View Folder), highlight message to be

moved, S (Save), CONTROL-T (To Folders), N/P until **tmp-mail** is highlighted, S (Select), RETURN (Accept).

17. Try giving these two commands:

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

Explain the differences between them.

18. Repeat exercise 8 using the file **phone.gz**, a compressed version of the list of names and phone numbers. Try to consider more than one approach to each question, and explain how you chose your answer.

You can either decompress the file using `gunzip`, giving the same commands as exercise 8 used once the file is decompressed, or use `zcat` and a pipe to display the results without creating an intermediate file as shown following:

```
$ zcat phone.gz | grep "Ace Electronics"
$ zcat phone.gz | sort
$ zcat phone.gz | uniq
```

Which technique you use makes a significant difference only if the **phone.gz** file is large, in which case it is an issue of what you are doing and a tradeoff between using more CPU (processor) time and less hard disk storage or vice versa.

When you are giving one of the commands one time only, using a pipe is more efficient. When you want to give more than one of the commands or want to give one of the commands repeatedly, it is more efficient to decompress the file one time using `gunzip` and then process it repeatedly with `grep`, `sort`, or `uniq`, assuming you have sufficient disk space. Finally, the most inefficient technique as far as disk space goes and the most efficient as far as CPU (and your) time goes is to put the output of `grep`, `sort`, or `uniq` in new files using a redirect output symbol as shown following:

```
$ zcat phone.gz | sort > phone.sort
```

19. Find existing files or create files that
- `gzip` compresses by more than 80 percent
 - `gzip` compresses by less than 10 percent
 - get larger when compressed with `gzip`

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

20. Some mailers, particularly older ones, are not able to handle binary files. Suppose that you are mailing someone a file that has been compressed with `gzip`, which produces a binary file, and you do not know what mailer the recipient is using. Refer to the man page on `uuencode`, which converts a binary file to ASCII. Learn about the utility and how to use it.

- a. Convert a compressed file to ASCII, using `uuencode`. Is the encoded file bigger or smaller than the compressed file? Explain.

If `uuencode` is not on your system, you can download it from rpmfind.net; it is part of the GNU `sharutils` package. Refer to “Installing, Upgrading, and Removing Packages” on page 930.

The following command converts the file `memo.gz` to ASCII using `uuencode`. The `.uuencode` filename extension is not required.

```
$ uuencode memo.gz > memo.gz.uuencode
```

The resulting ASCII file is larger than the original binary file because `uuencode` includes control information.

- b. Would it ever make sense to use `uuencode` on a file before compressing it? Explain.

No purpose is served by using `uuencode` to convert a binary file to ASCII before compressing it because compressing an ASCII file creates a binary file.