

Answers to Even-Numbered Exercises

1

from page 21

1. What is *free software*? List three characteristics of free software.

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

GNU/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. For more information, refer to “What Is So Good about GNU/Linux?” on page 8.

The source code for the operating system is readily available so that students can understand more easily how GNU/Linux works and can modify the code further to understand its operation and change the way it works. For more information, refer to “The Code Is Free” on page 6.

3. What are multiuser systems? Why are they successful?

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

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 GNU/Linux Operating System.

Linux is the name of an operating system *kernel* developed by Linus Torvalds and expanded and improved by thousands of people on the Internet.

Torvalds's kernel and GNU's tools work together as the GNU/Linux Operating System.

5. In what language is GNU/Linux written? What does the language have to do with the success of GNU/Linux?

6. What is a utility program?

A utility (program), sometimes referred to as a command, is a program that performs a task that is frequently related to the operating system. A utility is simpler than an application program although there is no clear line separating the two. GNU/Linux distributions, such as Red Hat, include many utilities. You can 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).

7. What is a shell? How does it work with the kernel? With the user?

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

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 executes the commands in the file, as though you had typed each command individually.

9. Why is the GNU/Linux filesystem referred to as *hierarchical*?

10. What is the difference between a multiprocessor and a multiprocessing system?

Multiprocessor (having more than one processor) refers to a system with two or more processors or CPUs.

Multiprocessing (supporting multiple processes) refers to a system that can process one or more tasks at a time.

11. Give an example of when you would want to use a multiprocessing system.

12. Approximately how many people wrote GNU/Linux? Why is this unique?

Many thousands of people contributed to the GNU/Linux Operating System using the Internet. This project is unique because such a project, using free software, had never been attempted before.

13. Who owns GNU/Linux? What are the key terms of the GNU General Public License?

14. Your system has a utility named `cut`. Find out what it does, and give an example of how you might use it.

Refer to page 1132 in Part III for more information on `cut`. Or, use the `man` or `info` utility to read about `cut`.