

## ANSWERS TO EXERCISES

---

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

“Free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer.”  
—Richard Stallman (page 2)

Four freedoms characterize free software ([www.gnu.org/philosophy/free-sw.html](http://www.gnu.org/philosophy/free-sw.html)):

- a. The freedom to run the program for any purpose (freedom 0).
- b. The freedom to study how the program works and to adapt it to your needs (freedom 1). Access to the source code is a precondition for this freedom.
- c. The freedom to redistribute copies so you can help your neighbor (freedom 2).
- d. The freedom to improve the program and to release your improvements to the public so that the entire community benefits (freedom 3). Access to the source code is a precondition for this freedom.

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.

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

Multiuser systems divide computer resources among multiple users, allowing for more efficient use of these resources. They are more cost-effective than single-user operating systems. Refer to “Linux Can Support Many Users” on page 12.

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](http://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.

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

Most of Linux is written in the C programming language. When written in a portable style, C programs can be moved from one platform (processor- or CPU-based system) to another by simply recompiling the code. Portability means that manufacturers have a ready supply of software, operating systems, and applications when they modify an existing platform or develop a new one.

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.

7. What is the difference between the terms *free software* and *open-source software*? Who coined each term?

The term *free software* is a philosophical and political label coined by the FSF (Free Software Foundation).

---

The term *open source* is more pragmatic, referring to the benefits of having a group of people work on and share software. It was created by a group of people who met in February 1998 and promoted by the OSI (Open Source Initiative).

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).

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

A shell is a command interpreter; it starts the program you call from a command line and passes your instructions (arguments) to the program. The shell is also a programming language; it can run files of commands, similar to DOS batch files, when you issue a single command to the shell.

When you use a shell in interactive mode (from a command line), the kernel accepts input from your terminal (by default) and passes it to the shell. It also takes output from the shell and displays it on the screen (by default).

The shell accepts a command from a user, starts execution of the program named in a command, and passes the command line to the program.

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`.)

11. Why is the Linux filesystem referred to as *hierarchical*?

The Linux filesystem is a hierarchy in which the root directory appears at the top of the system and branches come off the root, with each branch supporting one or more plain or directory files.

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.

13. Give an example of when you would want to use a multitasking system.

Multitasking is required in the following scenarios:

- a. You want to run two programs at the same time. For example, you want to print a report while you are working on something else.
- b. More than one person is using the system, each running a separate process.
- c. You want to run a graphical user interface (GUI). Each window you open runs a separate process.

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.

15. What are the key terms of the GNU General Public License?

The GPL says that you have the right to copy, modify, and redistribute the code covered by the agreement. However, when you redistribute the code, you must also distribute an equivalent license with the code, making the code and the license inseparable.

## ANSWERS TO EXERCISES

---

1. Briefly, what does the process of installing an operating system such as Linux involve?

When you install an operating system such as Linux, you copy operating system files from an installation medium to hard disk(s) on a system and set up configuration files so that Linux runs properly on the hardware.

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.

3. Would you set up a GUI on a server system? Why or why not?

Servers do not generally have GUIs. Typically you want to dedicate as many resources to the server as possible and few resources to anything not required by the server. A GUI consumes a lot of system resources and is not required by the server. Also, the presence of additional software on a server makes a system more vulnerable to attack.

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.

5. What is free space on a hard disk? What is a filesystem?

Free space on a hard disk is space that you can use for partitions. A filesystem is a data structure that resides in a partition.

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**.

7. Give two reasons why RAID cannot replace backups.

If the system experiences a catastrophic failure and the hard disks are destroyed or missing, RAID will be useless.

You cannot use RAID to replace a file when it is deleted by accident.

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.

9. What is SHA2? How does it work to ensure that an ISO image file you download is correct?

SHA2 (Secure Hash Algorithm 2) is a one-way hash function. A one-way function takes a variable-length message and produces a fixed-length hash (string of characters). When you run SHA2 against a file, it always produces the same hash. If the file is changed, SHA2 produces a different hash.

When Fedora/RHEL creates an ISO image file, it runs SHA2 against the file and publishes the resulting hash. After you download the file and run SHA2 against the downloaded copy of the file, if the resulting hash is the same as the one Fedora/RHEL published, the file is correct.