Chapter 16

Linux Operating System

At a Glance

Instructor's Manual Table of Contents

Lecture Notes

Overview

Linux is not UNIX. Linux was based on a version of UNIX but capitalized on the lessons learned over the previous twenty years of UNIX development. Linux has unique features that set it apart from its predecessor and make it a global force in operating systems development.

This chapter begins by presenting the history of the Linux operating system. It then discusses design goals. The roles of the Memory, Processor, Device, and File Managers in Linux are discussed. The chapter concludes with a discussion on the user interfaces in this operating system and various tools and features included in the operating system.

Learning Objectives

After completing this chapter, the student should be able to describe:

- The design goals for the Linux operating system
- The significance of using files to manipulate devices
- The differences between command-driven and menu-driven interfaces
- The roles of the Memory, Device, File, Processor, and Network Managers
- Some strengths and weaknesses of Linux

Teaching Tips

Overview

1. Provide students with an overview of the Linux operating system. Be sure to mention that it is POSIX-compliant. Discuss its major advantages: portability, availability though free source code, powerful GUI interfaces, and modularity.

2. Emphasize that Linux is a freely available operating system, which differentiates it from the operating systems discussed so far.

Teaching Tip	Refer to the following Web site for an overview of the Linux operating system: www.linux.org/info/

History

1. Provide students with a brief overview of the history of the Linux operating system. Point out that it was originally intended to be a limited capability operating system for the Intel 80386 chip, and its roots are in a miniature UNIX system called Minix.

2. Discuss the features of the original operating system. Point out that it was created to run a small microcomputer with the flexibility and functions found on expensive commercial operating systems. Note that because Linux was based on UNIX, it brought the speed, efficiency, and flexibility of UNIX to a small computer environment.

3. Discuss the open-source aspect of Linux and the implementation of updates.

4. Explain the original user interface mechanism (typed and cryptic commands). Discuss how today's users have an option to use either a command-driven interface (Terminal mode) or a graphical user interface (GUI), greatly expanding the usability of the operating system.

5. Discuss the importance of Red Hat Linux in terms of their support and distribution of the product until 2003. Explain the split in 2003 and its effect on the Linux operating system. Specifically, Red Hat Linux is now distributed freely through Fedora, and the Red Hat Enterprise Linux is a commercial version.

6. Explain the evolution of Linux using Table 16.1 on page 497-498 as a guide. Include in your discussion the historical roots from UNIX, its features, and modifications.

7. Discuss the advantage of distributing Linux under GNU General Public License.

8. Discuss the three design goals of Linux (modularity, simplicity, and portability), and how they are achieved. Point out that Linux supports numerous standard utilities and many functions.

9. Review some of the functions supported in Linux using Table 16.2 on pages 498-499 as a guide.

10. Point out that Linux conforms to the specifications for Portable Operating System Interface for Computer Environments (POSIX), which is an IEEE standard that defines operating system interfaces to enhance the portability of programs from one operating system to another.

Teaching Tip	Refer to the following Web sites for more information on IEEE POSIX: • http://standards.ieee.org/regauth/posix/index.html • http://posixcertified.ieee.org

Teaching Tip	Refer to the following Web site for information about the GNU General Public License: www.gnu.org/licenses/gpl.html

Teaching Tip	Refer to the following Web site for an article on Linus Torvalds, the writer of the original Linux operating system: www.howstuffworks.com/framed.htm?parent=question246.htm&url=http://www.metroactive.com/papers/metro/05.08.97/cover/linus-9719.html

Memory Management

1. Discuss how Linux performs space allocation among the kernel and the executing processes.

2. Explain what happens when a process begins execution in terms of memory allocation using segments. Point out that Linux can change the size of process data segments using system calls.

3. Discuss how Linux offers memory protection depending on the type of information stored in each region belonging to the address space of a process.

4. Discuss what happens when a process requests pages and how Linux loads them into memory. Be sure to mention that Linux uses the Least Recently Used (LRU) algorithm. Point out that Linux maintains a dynamically managed area in memory and a page cache where new and old pages may be inserted and deleted.

5. Discuss how Linux keeps track of free and busy pages using system page tables.

6. Discuss how virtual memory in Linux is managed using a multiple-level table hierarchy, accommodating both 64- and 32-bit architectures and demand paging.

7. Explain the four fields that make up the virtual address using Table 16.3 on page 500 as an example. Use Figure 16.1 on page 501 to demonstrate Linux virtual memory management.

8. Explain the Buddy algorithm. Discuss how it manages the grouping and splitting of equal-sized page frames to give more contiguous space to a job. Use the example on page 501 along with Figure 16.2 on page 502 to clarify main memory management.

9. Discuss the page replacement algorithm. Point out that it is an expanded version of the clock page replacement policy. Explain how it works in Linux. Note that Linux does not use a single reference bit. Instead, Linux uses an eight-bit byte to keep track of a page's activity, which is referred to as its age.

Teaching Tip	Note that Linux shows added flexibility with swap devices because, if necessary, Linux can deactivate them without having to reboot the system.

Teaching Tip	Refer to the following Web site for an article discussing an introductory-level tour of Linux memory models and management: www.ibm.com/developerworks/linux/library/l-memmod

Processor Management

1. Provide students with a brief introduction to processor management in the Linux operating system. Point out that Linux uses the same parent-child process management design found in UNIX. Be sure to cover the concept of "personality" that is supported in Linux.

Organization of Table of Processes

1. Describe how each process is referenced by a descriptor. Point out the characteristics of a descriptor and its use by the kernel.

Process Synchronization

1. Explain wait queues and semaphores, which are used to synchronize two processes with each other.

2. Describe a wait queue and the two problems it solves.

3. Describe a semaphore. Explain the three fields in a semaphore.

Process Management

1. Provide students with a brief introduction to process management in the Linux operating system. Point out the three scheduling types for real-time and normal processes.

2. Use Table 16.4 on page 503 to explain how the combination of type along with priority is used by the scheduler to determine scheduling policy for processes in the READY queue.

3. Explain how the scheduler handles each of the three types.

Teaching Tip	Refer to the following Web site for an article discussing Linux Process Management: www.2000trainers.com/linux/linux-process-manage

Quick Quiz 1

1. Which of the following describes the problem(s) that wait queues resolve in process synchronization? (Choose all that apply.)
   a. Mutual exclusion
   b. Disconnection
   c. Producers and consumers
   d. Overwriting sections of memory
   Answer: a and c

2. Linux uses the _____ page replacement algorithm.
   a. MRU
   b. LRU
   c. FCFS
   d. FIFO
   Answer: b

3. (True or False) For general use, Linux is a free operating system.
   Answer: True

Device Management

1. Discuss the important features of device management in Linux operating systems. Emphasize that Linux is device independent and the device drivers supervise the transmission of data between main memory and the peripheral unit.

2. Explain that devices are assigned a name and a descriptor. Use Figure 16.3 on page 504 to explain how the descriptor further identifies the device and how they are stored in the directory.

3. Describe how Linux provides a comprehensive collection of device drivers. Note that if a driver required is not available, it may be obtained from another source and installed separately or manually written by a skilled programmer.

Device Classifications

1. Explain how Linux identifies each device by a major number and a minor number. Describe how both are used.

2. Discuss the Configuration Table available for each class of driver and the significance of the table. Note that it is an important feature of the operating system because it allows systems programmers the ability to create new device drivers quickly to accommodate differently configured systems.

Device Drivers

1. Provide students with a brief overview of device drivers in Linux. Point out that Linux supports the standard classes of drivers as defined in UNIX while allowing new device classes to support new technology.

2. Describe the nature of device classes, noting that they are not rigid. Point out why programming large complex drivers is discouraged by presenting two reasons against such programming.

3. Note that a notable feature of Linux is that it allows device drivers to be created on the fly while the system is up and running.

Teaching Tip	Refer to the following Web site for a tutorial on writing Linux device drivers: www.networkcomputing.com/unixworld/tutorial/010/010.txt.html

Device Classes

1. Note the three standard classes of devices supported in Linux, using Figure 16.4 on page 506 as a guide.

2. Explain character devices. Discuss how they work, noting the different ways to access them, how they are implemented, and how they are accessed.

3. Explain block devices. Point out the difference in a block device from a character device (can host a file system). Note their similarity to character devices in terms of appearing to be ordinary files, with the exception that the block drivers can access a file system in connection with the device, something not possible with the char device.

4. Explain network interfaces. Point out the dissimilarity between block devices and character devices. Discuss how each system device is handled by a device driver that is, in turn, under the direction of a subsystem of Linux.

5. Discuss the two common functions, open and release. Explain the purpose of each operation.

File Management

Data Structures

1. Discuss how all Linux files are organized in directories that are connected to each other in a treelike structure.

2. Point out that Linux specifies five types of files. Use Table 16.5 on page 508 to illustrate these types.

Filename Conventions

1. Discuss the general characteristics for filenames in Linux. Be sure to mention that filenames in Linux are case sensitive.

2. Describe the hierarchical tree file structure supported in Linux, using the examples presented on pages 508 to 509 and shown in Figure 16.5 on page 508 as a guide.

3. Explain the rules that apply to all path names, using the numbered points on page 509 as a guide.

4. Describe the Virtual File System layer in terms of purpose and functionality. Discuss the need for the kernel software layer to allow processes to access files in a consistent manner while maintaining an interface between system calls related to files and the file management code.

Directory Listings

1. Discuss how directory listings can be created by using Terminal mode with typed commands (ls or ls -l) or by using the GUI desktop.

2. Describe what is contained in a typical listing (name of the file or directory, its size, and the date and time of modification). Note that information about file permissions may be added using the View option on the menu bar.

3. Use Figure 16.6 on page 510 to illustrate a sample list of files stored in a directory including file permissions.

4. Discuss the Permissions column, noting that it shows a code with the file’s type and access privileges. Use Figure 16.7 on page 510 to illustrate this pictorially.

5. Describe the content and meaning of each set of characters in the permissions column.

6. Explain to the students how to change the security for a file. The owner (and only the owner) can open the properties for the file to be protected by choosing File-Properties from the File menu and then clicking on the Permissions tab to choose the appropriate access for the owner, group, and others, respectively.

7. Use Figure 16.8 on page 511 in the text to illustrate the permission tab layout.

Teaching Tip	Refer to the following Web site for a tutorial on the Linux File Manager: www.reallylinux.com/docs/windowstolinux.shtml

Teaching Tip	Refer to the following Web site for an article reviewing several Linux File Managers: www.linux.com/feature/113952

User Interface

1. Discuss the user interface available in early versions of Linux. Note that a user had to type commands and needed a thorough knowledge of valid commands.

2. Discuss the user interface available in today's versions of Linux. Note that most current versions include powerful and intuitive GUI desktops.

3. Emphasize that it is important to realize that users can still use Terminal mode to type commands that are very similar to those used for UNIX.

Command-Driven Interfaces

1. Describe the general syntax for typed commands. Explain the three parts of the command syntax (command arguments filename) in more detail. Note that commands must be in the correct case (usually lowercase letters).

2. Use Table 16.6 on page 512 to illustrate sample commands. Discuss how commands can be abbreviated and combined.

3. Discuss how commands are interpreted and executed by the shell. Provide an example of a shell using the Bash shell, which is one of the two most widely used shells.

4. Explain that a shell is technically known as the command interpreter. Emphasize the key role it plays in the coordination and combination of system programs.

Graphical User Interfaces

1. Discuss how most Linux operating systems are delivered with multiple graphical user interfaces. Note that these are often free of charge. Describe the advantage this offers by allowing the end users to choose the GUI that best meets their needs or those of the organization.

2. Discuss the flexibility provided by the different GUIs. Explain that these GUIs and their flexibility have spurred the acceptance of Linux and have helped it become more competitive.

3. Discuss the sophisticated Windows-compatible word processors, spreadsheet, and presentation application tools available in Linux. Point out that these tools make it possible for Linux users to read and write documents that are generated, or read, by colleagues using proprietary software from competing operating systems distributors. Note that this is one factor spurring the popularity of Linux.

Teaching Tip	Refer to the following Web site for an article discussing the "Top 3 Free Office Suites for Unix/Linux": http://linux.about.com/od/softoffice/tp/tp_officesuite.htm

System Monitor

1. Discuss the information available using the System Monitor window.

2. Use Figure 16.9 on page 513 to illustrate a system monitor and sample output.

Service Settings

1. Discuss the variety of services available to help manage the system. Use Figure 16.10 on page 513 as a guide to illustrate a sample service settings window.

2. Explain that options may vary from one system to another, and it is best to verify details in the system manual for system specific information.

System Logs

1. Discuss how system logs provide a detailed description of activity on the system. Point out the types of activities recorded in system logs (tracking system malfunction, firewall failure, disabled device).

2. Mention the location for these log files on some Linux operating systems (the /var/log directory).

3. Use Figure 16.11 on page 515 to illustrate a sample log viewer.

4. Explain that there are numerous log files available for review. Use Table 16.7 on page 515 to illustrate sample Linux log files.

Keyboard Shortcuts

1. Note that one of the advantages of keyboard shortcuts is that they allow users to switch easily from one task to another.

2. Use Figure 16.12 on page 512 to illustrate some shortcuts. Point out that many Linux shortcuts are identical to those commonly used on Windows operating systems. Note that this eases operating system transition.

3. Provide an example such as CTRL-V to illustrate the commonality of shortcuts.

Teaching Tip	Refer to the following Web site for an article discussing Linux administration tools: www.linuxjournal.com/article/5918

Quick Quiz 2

1. In Linux, programming large complex drivers is discouraged for the following two reasons. (Choose two.)
   a. It is not possible to manually code and install drivers for the Linux operating system.
   b. The shared code aspect of Linux suggests a need for a wide demand for several simple drivers rather than a single complex one.
   c. Modular code is better able to support the goals of system scalability and extendibility in Linux.
   d. Coding changes to Linux by non-system developers is not encouraged in the Linux environment.

   Answer: b and c

2. (True or False) Linux provides only one type of user interface: GUIs.
   Answer: False

3. The _____ provide(s) a detailed description of activity on the system. (Choose the one best answer.)
   a. System Monitor
   b. service settings
   c. system log
   d. keyboard shortcuts

   Answer: c

Teaching Tip	Refer to the following Web site for a tutorial article for new users of the Linux desktop: http://linux.about.com/od/linux101/a/desktop.htm

Teaching Tip	Refer to the following Web site for links to the vendor sites for 100 distributions of Linux: http://linux.about.com/od/embedded/l/bldist_100az.htm

Class Discussion Topics

1. Have students discuss device management in the Linux operating systems. How is it different from device management in Windows operating systems? Ask students to list all the advanced key features of device management in Linux operating systems.

2. Ask students if any of them has worked with Linux in the past, and if so, how they felt about the experience. What were some of the challenges that they faced in attempting to install and configure Linux? Do they think that Linux will be a long-term and viable alternative to Windows and UNIX? Why or why not?

Additional Projects

1. Have students research online and compile a chart illustrating the major advantages and disadvantages of the Windows and Linux operating systems. The chart should be a guide for users who may consider making a change from Windows to Linux or from Linux to Windows.

2. Have students research online and create a list of what they consider to be the best online Linux resources available, along with the types of information each site provides. Once completed, compile the students' lists into a master list to be distributed to the class.

3. Have students research online to compile a list of the advanced features of the Red Hat Global File System (GFS).

Additional Resources

1. IBM.com:
   www.ibm.com

2. Linux.org:
   www.linux.org

3. Linux-directory.com:
   www.linux-directory.com/

4. Red Hat Linux:
   www.redhat.com

5. Linux Format Magazine:
   www.linuxformat.co.uk

6. Kernel.org:
   www.kernel.org

7. The Linux Journal:
   www.linuxjournal.com

8. GNU General Public License:
   www.gnu.org/licenses/gpl.html

9. IEEE:
   www.ieee.org

Key Terms

- **Argument:** in a command-driven operating system, a value or option placed in the command that modifies how the command is to be carried out.
- **Buddy algorithm:** a memory allocation technique that divides memory into halves to try to give a best-fit and to fill memory requests as suitably as possible.
- **Child process:** subordinate processes that are controlled by a parent process.
- **Clock page replacement policy:** a variation of the LRU policy that removes from main memory the pages that show the least amount of activity during recent clock cycles.
- **Command:** a directive to a computer program acting as an interpreter of some kind to perform a specific action.
- **Command-driven interface:** an interface that accepts typed commands, one line at a time, from the user. It is also called command line interface and contrasts with a menu driven interface.
- **CPU-bound:** a job that will perform a great deal of nonstop processing before issuing an interrupt. A CPU-bound job can tie up the CPU for long periods of time.
- **Device driver:** a device-specific program module that handles the interrupts and controls a particular type of device.
- **Device independent:** programs that can work on a variety of computers and with a variety of devices.
- **Directory:** a logical storage unit that contains files.
- **Graphical user interface (GUI):** allows the user to activate operating system commands by clicking on icons or symbols using a pointing device such as a mouse. It is also called a menu-driven interface.
- **Kernel:** the part of the operating system that resides in main memory at all times and performs the most essential tasks, such as managing memory and handling disk input and output.
- **Menu-driven interface:** an interface that accepts instructions that users choose from a menu of valid choices. It is also called a graphical user interface and contrasts with a command-driven interface.
- **Parent process:** a job that controls one or more child processes, which are subordinate to it.
- **Portable Operating System Interface for Computer Environments (POSIX):** a set of IEEE standards that defines the standard user and programming interfaces for operating systems so developers can port programs from one operating system to another.