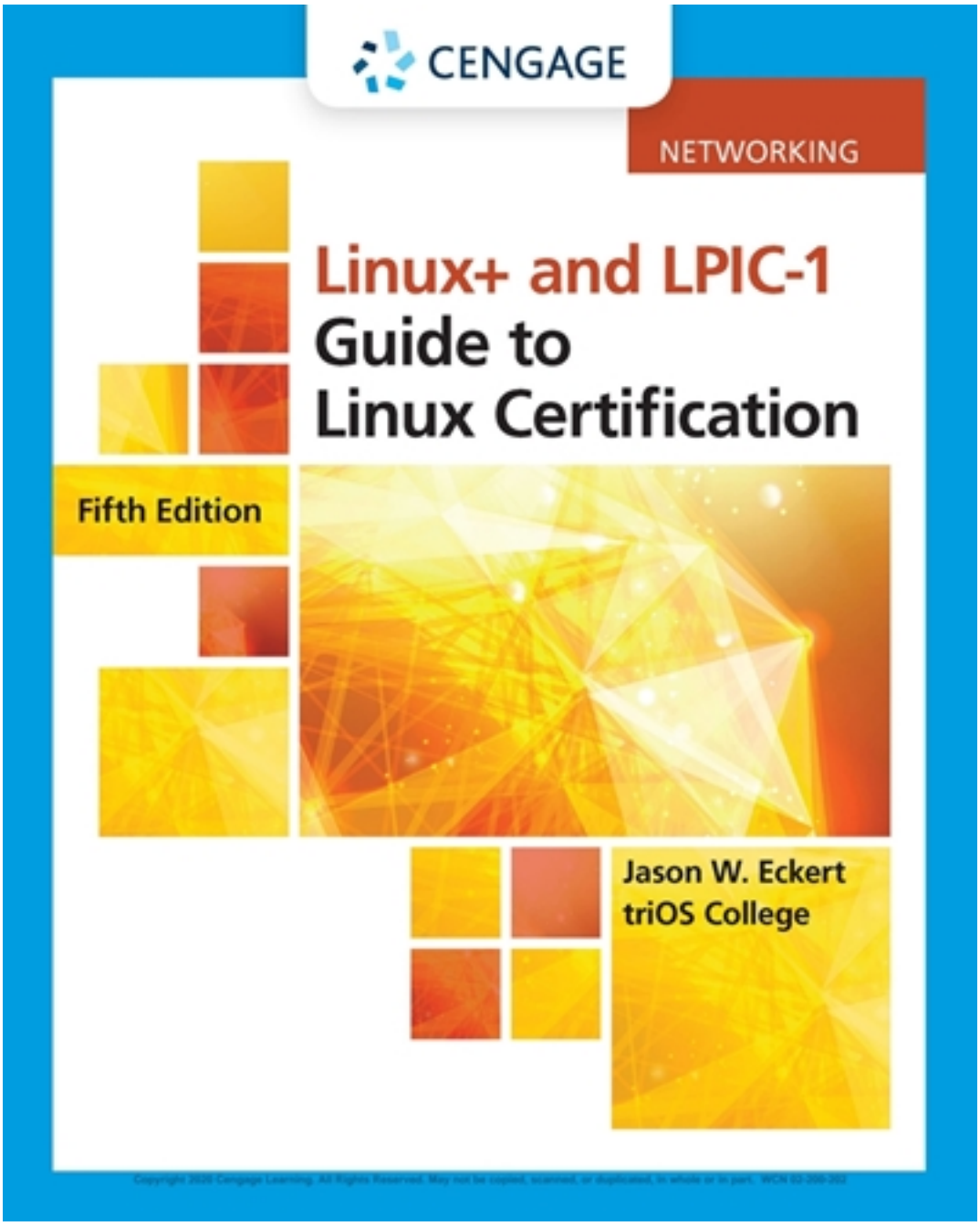
Solutions for Linux and LPIC-1 Guide to Linux Certification 5th Edition by Eckert

CLICK HERE TO ACCESS COMPLETE Solutions



Solutions

Chapter 1

Introduction to Linux

At a Glance

Instructor's Manual Table of Contents

- Overview
- Objectives
- Teaching Tips
- Quick Quizzes
- Class Discussion Topics
- Additional Projects
- Additional Resources
- Key Terms
- Technical Notes for Hands-On Projects

Lecture Notes

Overview

Linux technical expertise is essential in today's computer workplace as more and more companies switch to Linux to meet their computing needs. Thus, it is important to understand how Linux can be used, what benefits Linux offers to a company, and how Linux has developed and continues to develop. In the first half of this chapter, you learn about operating system terminology and features of the Linux operating system, as well as the history and development of Linux. Later in this chapter, you learn about the various types of Linux distributions and about the situations in which Linux is used.

Chapter Objectives

After completing this chapter, your students will be able to:

- Explain the purpose of an operating system
- Outline the key features of the Linux operating system
- Describe the origins of the Linux operating systems
- Identify the characteristics of various Linux distributions and where to find them
- Explain the common uses of Linux in industry today

Teaching Tips

Operating Systems

1. Provide students with an overview of the common hardware components found in computers, explaining the basic function of each. Be sure to concentrate on the PC architecture primarily, but outline how different systems differ in terms of common components.

2. Explain the concept of a software program, outlining the difference between a program and a process.

3. Outline the purpose of an operating system (OS); explain the interaction that takes place between an OS, applications, and computer hardware. Explain how users interact with each of these three elements.

4. Explain why an operating system uses device drivers.

5. Provide students with an overview of the different types of system interfaces commonly found in OSs and how these environments can be accessed. Be sure to discuss user

interfaces and graphical user interfaces (GUI). Examples suitable for brief discussion are printer drivers and command prompt user interface.

6. Explain the concept of system services, outlining some of the common services provided by client and server systems. Examples suitable for brief discussion include file and print services, network services such as Domain Name Space (DNS) and Dynamic Host Configuration Protocol (DHCP).

Teaching Tip	As class begins, reassure students that you are here to help them understand technology. Many less technically oriented students will feel intimidated by this course. Get off to a good start!

The Linux Operating System

1. Provide students with a brief introduction to the Linux OS, outlining how the system can be used to provide both client and server services to many simultaneous users.

2. Explain the concepts of a multi-user and multi-tasking system as it relates to an OS like Linux.

Versions of the Linux Operating System

1. Provide students with an overview of the purpose and primary functions of the Linux kernel.

2. Explain the Linux kernel development process, emphasizing the importance of understanding how different versions support different needs.

3. Emphasize that to take advantage of new technologies or to fix problems (also known as bugs) related to your computer's hardware, sometimes the kernel must be upgraded after installing Linux.

4. Explain why a good understanding of a system's hardware is important when determining which kernel to use.

5. Mention that in some cases, updates in the form of a kernel module or a kernel patch can be used to provide or fix hardware supported by the kernel.

Teaching Tip	Introduce students to the Linux kernel Web site at https://www.kernel.org.

Identifying Kernel Versions

1. Provide students with an overview of the Linux kernel version numbering system, outlining the differences among major, minor, and revision numbers.

2. Explain the difference between development and production kernel versions. Be sure that students recognize the importance of using only production kernel versions on all systems but those used for testing purposes.

3. Use Table 1-1 to provide students with a basic overview of the Linux kernel version history, outlining how the process now moves much faster due to the widespread business support and larger development community that has grown with the popularity of Linux as a viable enterprise computing alternative.

Licensing Linux

1. Provide students with an overview of the concept of open source software (OSS), explaining how this development process differs from closed source alternatives.

Teaching Tip	Find the complete open source definition at: https://opensource.org.

2. Explain the concept of source code making sure students understand how source code ends up as code a computer's processor can understand and execute.

3. Explain why software developers are able to write Linux source code in many different programming languages.

Teaching Tip	For a short list of user voted "Best Programming Languages", see: https://www.linuxjournal.com/content/best-programming-language

4. Discuss the following advantages of the OSS way of developing software.
   - Software is developed rapidly through widespread collaboration
   - Software bugs (errors) are noted and promptly fixed
   - Software features evolve quickly, based on users' needs
   - The perceived value of the software increases because it is based on usefulness and not on price

5. Use Table 1-2 to explain the differences among open source, closed source, freeware, and shareware software. Provide common examples of each, using popular software packages as examples.

Types of Open Source Licenses

1. Provide students with a basic overview of the GNU Public License (GPL) and the Free Software Foundation (FSF), explaining that the GPL is the most popular licensing model for Linux software used today. Provide students with examples of software packages released under the GPL.

Teaching Tip	For more information about the GPL, visit: https://www.gnu.org.

2. Explain the concept of the Artistic License, outlining how it differs from the GPL. Provide students with examples of software released under an Artistic License.

Types of Closed Source Licenses

1. Explain the various types of closed source licenses that exist, including examples of how freeware and shareware fit into this category.

Linux Advantages

1. Point out that Linux is the fastest growing operating system released to date and outline some of the reasons for its phenomenal growth since 1998. Some ideas to consider include developer support, its open source nature, large company support and investment, and so on.

2. Provide students with an overview of the primary advantages that can be derived from using Linux as an OS alternative.

Risk Reduction

1. Explain why companies are exposed to costly and time-consuming software update risks when investing in non-OSS software applications used for mission-critical tasks.

2. Outline some of the advantages afforded to a company using OSS products such as Linux for mission-critical tasks. Important considerations include the ability to take the source code, add features to it, and maintain it themselves provided the source code was redistributed free of charge. Also, most OSS does not retire after a short period of time because collaborative open source development results in constant software improvement geared to the needs of the users.

Meeting Business Needs

1. Outline some of the more common software types available for Linux.

2. Explain how the use of UNIX in the business world has affected the growth of Linux and businesses ease of transition to Linux.

3. Explain how companies can take advantage of several educational resources and certification exams for various Linux skill levels.

4. Point out that Linux provides support for most programming languages.

Stability and Security

1. Compare the relative advantages and disadvantages of updating or patching open- versus closed-source software. Explain some of the reasons why a closed source vendor might be reluctant to announce that a particular security flaw exists in their software.

2. Discuss some of the reasons why Linux is strong in security compared to other closed sourced operating systems. Point out that the number of viruses associated with Windows is so much higher than the number associated with Linux.

3. Explain the primary reason why most desktop and server Linux systems run antivirus and antimalware software today even though the number of viruses that can affect Linux is exceedingly low.

Teaching Tip	To find a list of recent computer viruses, visit: https://securelist.com.

Flexibility for Different Hardware Platforms

1. Discuss the different hardware platforms on which Linux is capable of running. Point out which of these platforms are most popular, and that certain distributions only support certain platforms. Stress the differences in size and work capacity between the various hardware platforms.

2. Explain the concept of an embedded system and how this differs from a traditional OS installation. Point out this focus on mobile and embedded devices will become more important in the future as the need for new functionality increases.

Ease of Customization

1. Explain how the Linux kernel can be customized to include and exclude packages as necessary. Outline the benefits of a smaller kernel, including how this increases performance and can help to avoid various security issues associated with many applications and services being installed.

2. Mention that most Linux configurations offer hundreds of small utilities, which, when combined with Shell or PERL programming, can quickly make new programs that meet many business needs.

Ease of Obtaining Support

1. Describe some of the different methods of obtaining Linux support, including frequently asked questions (FAQs), HOWTO documents, the Linux Documentation Project (LDP) Internet newsgroups, Linux-focused Facebook groups and website forums, telephone service, Linux User Groups (LUGs), and so forth.

2. If one or more exist, provide students with the URLs for Linux User Groups (LUGs) in your geographic area.

Cost Reduction

1. Provide students with an overview of the different cost advantages associated with running Linux rather than other closed source alternatives like UNIX or Windows.

2. Outline some of the ways in which using Linux can lead to increased costs, such as user training and support.

3. Explain the concept of total cost of ownership (TCO), outlining the different elements involved in its calculation. Use Table 1-3 in the text as a guide.

Quick Quiz 1

1. What term is used to describe a running program on Linux?
   a. Application
   b. Process
   c. Runtime
   d. Project
   Answer: B

2. What is represented by the second number in the Linux kernel version 2.3.4?
   a. Minor number
   b. Major number
   c. Production kernel
   d. Development kernel
   Answers: A

3. Under which licensing model is Linux made available?

a. GPL
b. Freeware
c. Shareware
d. Artistic License
Answer: A

4. Which of the following describes a type of closed source license which is distributed free of charge, yet after a certain number of hours of usage or to gain certain features of the program, payment is required?
a. Artistic License
b. Shareware
c. Free Software Foundation
d. Freeware
Answer: B

5. A task-specific instruction guide to performing a wide variety of tasks that is freely available from the Linux Documentation Project is known as which of the following?
a. Frequently asked questions (FAQs)
b. man pages
c. HOWTO documents
d. Linux User Groups (LUGs)
Answer: C

The History of Linux

1. Provide students with a brief overview of the timelines associated with the development of the UNIX and Linux OSs. Use Figure 1-4 in your discussion.

UNIX

1. Provide students with a brief overview of the history of UNIX development, outlining the various hardware platforms (such as the PDP-7) on which UNIX ran.

2. Explain the relative importance of the C programming language as it has related to the development of application software.

3. Outline the ways in which the various distributions of UNIX were formed and the impact that this ultimately had on the development of computing platforms worldwide.

The Hacker Culture

1. Provide students with an overview of the concept of a hacker and what the word means in a pure sense. Discuss the difference between a cracker and a hacker.

2. Explain the objective of the GNU Project, outlining how it has become a force in the open source computing movement.

3. Explain the primary ideas behind each of the hacker culture bullet points listed on Page 19 of the text. Outline some of the ways in which these points are true, as well as the way that many of the original concepts of a hacker have changed over time.

Teaching Tip	A description of FSF and GNU can be found online at: https://www.gnu.org.

Linux

1. Provide students with an overview of the way in which the work of Linus Torvalds developed into the Linux used today. Stress the fact that while Linux was originally developed by one person, its strength lies in the collaborative work of many people worldwide.

2. Explain the concept of a Linux distribution and identify some of the well-known distributions such as Red Hat, openSUSE, Debian, Ubuntu, Gentoo, Linux Mint, and Arch.

3. Explain why there was a shift in Linux development during the 2000s to support the larger computing environments and mobile devices.

4. Remind student that Linux is a by-product of OSS development and because Linux itself is currently very well developed, even more application development can be expected from the OSS community in the next decade.

Linux Distributions

1. Briefly explain the important differences Linux distributions can have despite being essentially the same under the surface.

2. Explain the concept of a distribution kernel pointing out why they are used and how they are identified numerically.

3. Describe an example of when a Linux distribution that includes many specialized tools might not contain a GUI.

4. Explain the concept of an X window as the core component of a Graphical User Interface (GUI), and the need for the X window to interact with a windows manager and a desktop environment in order to create a full GUI environment.

5. Briefly introduce the two main Linux graphical user interface (GUI) environments: the GNU Object Model Environment (GNOME) and the K Desktop Environment (KDE).

Outline some of the main differences between these two desktops including their respective toolkits.

6. Provide students with a brief demonstration of both the GNOME and KDE GUI environments, pointing out some of the major and more subtle differences between the two. Use Figure 1-5 and Figure 1-6 in your discussion.

7. Explain the concept of a package manager and discuss examples like the Red Hat Package Manager (RPM) format used by Red Hat that have become so popular. Explain why someone might want to download software in tarball format. Discuss some of the issues associated with tarballs and why package managers are replacing them.

8. Explain why there are over 500 registered Linux distributions. Outline the hardware platforms supported by common Linux distributions, using Table 1-4 in the text as a guide. Review some of the reasons why a Linux vendor might choose to support some platforms and not others.

    a. Remind students that, for the most part, all Linux distributions are functionally similar, providing a Linux kernel and many of the same OSS packages. Outline that the main functional differences are a product of vendor customization, ease-of-use features such as installer programs, different software installation methods and tools, and so forth.

Common Uses of Linux

1. Reiterate that Linux is very versatile in providing a variety of easily configurable services. Provide students with a brief overview of the differences between workstation services and server services, providing examples of each.

2. Outline the Linux configurations commonly used today.

Internet Servers

1. Provide students with an overview of the most common services Linux provides on Internet servers. Use the bulleted list on Pages 26-27 as a guide.

Web Services

1. Explain that out of all the available Linux Internet tools and services, the most popular is the Internet browser, which can connect client computers to the community of servers known as the World Wide Web (WWW) or an individual server known as a Web server. Provide students with a brief demonstration of the Apache Web server.

2. Briefly discuss the Hypertext Transfer Protocol (HTTP) and how a Web server translates a request for a Web resource into a reply.

3. Explain that Web servers can process programs known as Common Gateway Interface (CGI) scripts and provide secure connections such as Secure Sockets layer (SSL) or Transport Layer Security (TLS). Discuss the importance of CGI scripts and the ability to provide secure connections.

DNS Services

1. Explain the importance of having each computer on a network identified uniquely using IP addresses.

2. Explain the concept of using a fully qualified domain name (FQDN) to simplify the complexity that can surround IP addresses and demonstrate its use to the class in a browser window.

3. Provide students with an overview of the DNS and the function that it provides for name resolution on the Internet. Point out that a Domain Name Service known as BIND (Berkeley Internet Name Daemon) ships with many distributions of Linux.

Teaching Tip	You can find the latest version of BIND at the Internet Systems Consortium website: https://www.isc.org

DHCP Services

1. Describe the importance of using a Dynamic Host Configuration Protocol (DHCP) server to configure workstations for accessing the Internet.

2. Any Linux computer can function as a DHCP server by adding and configuring the DHCP daemon, dhcpd.

Time Services

1. Point out that operating systems like Linux need the correct time to function properly.

2. Explain how a Linux system can obtain time information from an NTP server or provide time information to other systems using NTP via the NTP daemon (ntpd) or Chrony NTP daemon (chronyd).

Mail Services

1. Discuss the purpose of mail services and how they relate to both sending e-mail and accessing messages by users.

2. Explain the difference between a mail transfer agent (MTA) such as Sendmail and a mail delivery agent (MDA). Mention the MTAs that are freely available for Linux, including

sendmail, postfix, and exim. Mention that Linux provides several of these services with getmail and mpop being two of the most common.

3. Outline the concept of a mail user agent (MUA). Mention that common MUAs available for Linux include mutt, Alpine, Mozilla Thunderbird, and Claws Mail.

FTP Services

1. Provide students with an overview of the File Transfer Protocol (FTP) service and the function that it provides on a network.

2. Explain the difference between an FTP client and an FTP server, stressing the purpose of each.

3. Explain the function of an anonymous FTP server

4. Explain that most operating systems, such as Linux, UNIX, Microsoft Windows, and macOS, are distributed with an FTP client program, making it easy for users to connect to these FTP servers.

Authentication Services

1. Describe authentication and emphasize its importance in computer security.

2. Explain how an authentication service like the Kerberos protocol assist with the authentication process on a large network. Summarize the Kerberos process.

3. Provide relevant Kerberos-based authentication examples such as Microsoft Active Directory and the Apache Directory used on Linux servers.

Teaching Tip	You can learn more about Apache Directory at https://directory.apache.org.

Certificate Services

1. Explain how encryption algorithms protect data before it is transmitted on the network.

2. Compare symmetric encryption to asymmetric encryption explaining how they are different. Discuss why asymmetric encryption requires a public key and a private key.

3. Explain how private keys can be used to authenticate a messages. Explain that a message that has been encrypted using a private key is called a digital signature.

4. Describe the responsibility of the Certification Authority (CA). Explain that a public key that has been digitally signed by a CA is called a certificate. Mention that an organization that installs one or more CAs is said to have a Public Key Infrastructure (PKI).

Teaching Tip	You can learn more about Certificate Authorities at: https://searchsecurity.techtarget.com/definition/certificate-authority.

Routing Services

1. Briefly explain the concept of routing as it relates to the functionality of the Internet. Point out that companies can also use routers to connect their internal networks.

2. Briefly introduce the concept of a routing protocol.

3. Explain why Linux provides support for routing and is easily customizable and many Linux distributions provide routing capabilities.

Firewall and Proxy Services

1. Outline the purpose of a network firewall, explaining how this service is used to control which traffic can enter or exit a network interface. Discuss the placement of firewalls to protect a network from Internet users, as well as to control access to network services on individual systems.

2. Explain that Linux has firewall support built directly into the kernel. Mention that utilities such as ipchains and netfilter/iptables, included in most distributions, can be used to configure rules necessary to make a system act as a firewall.

Teaching Tip	You can find out more about using netfilter/iptables to configure Linux firewalls on the Internet at: https://netfilter.org/.

3. Outline how a proxy server is used to act as an intermediary between client OSs and a network like the Internet. Discuss how a proxy server caches Web Pages and associated information to speed up Internet access.

4. Briefly outline the purpose of Network Address Translation (NAT).

5. Briefly introduce students to Squid and outline some of its features.

Advanced Security Service

1. Explain what security appliances are and how they are used to provide security between an organization's network and the Internet.

2. Review the services provided by Linux-based commercial security appliances and security appliance software suites for Linux:
   - Malware and virus filtering
   - Spam filtering
   - Bot protection
   - Intrusion detection
   - Advanced traffic throttling
   - Virtual Private Network (VPN) functionality
   - Centralized event logging for network devices using Simple Network
   - Management Protocol (SNMP)
   - Security Information and Event Management (SIEM) for centralized network and security monitoring

Teaching Tip	Be sure to inform students that Security appliances that provide multiple security functions are often called Next Generation Firewall (NGFW) or Unified Threat Management (UTM) appliances. More information on UTM may be found at https://usa.kaspersky.com/resource-center/definitions/utm

File and Print Services

1. Provide an overview of implementing central storage using a network server. Explain why Linux is well suited to the task of centrally sharing resources.

2. Explain the fact that UNIX and Linux systems rely on network file systems (NFSs) for access to file and print servers.

3. Briefly outline the purpose of Samba in providing Windows clients with access to resources on Linux-based servers.

Application Servers

1. Discuss the functionality of an application server, and of its interaction with the user and with databases. Describe a database front-end/back-end relationship.

2. Explain the advantages of centralizing their key software elements on Internet application servers. Introduce students to the concept of a database management system (DBMS).

3. Point out that there are several free open source DBMS programs and tools available to facilitate creating, managing, and retrieving data from a database as well as interacting with closed source databases such as those from Microsoft and Oracle.

4. Emphasize that the most popular and widely used DBMSs available for Linux today are PostgreSQL, MySQL (My Structured Query Language), and MariaDB (based on MySQL). Discuss the advantages of using these software applications.

5. Point out that application servers can provide management functionality, allowing access and administration from anywhere in the world via the Internet.

Teaching Tip	Learn more about MySQL at: https://www.mysql.com.

Cloud Systems

1. Explain that, to most people, the cloud is another term for the Internet.

2. Describe the advantages of using cloud servers. Explain that a private data center that is accessible to other computers across the Internet is often called a private cloud and the organization is referred to as a cloud provider.

3. Discuss the three main approaches to hosting data and services within the cloud: SaaS, PaaS, and IaaS.

4. Describe cloud platform software. Mention that OpenStack is one of the most popular open source cloud platform software suites and is supported by nearly all major Linux distributions.

Teaching Tip	Learn more about OpenStack at: https://www.openstack.org

5. Explain why Linux is often used to implement SaaS, PaaS, and IaaS and is hosted within a virtualized IaaS cloud platform.

Teaching Tip	Learn more about Red Hat cloud computing at: https://www.redhat.com/en/topics/cloud

Supercomputers

1. Provide students with an overview of supercomputers, including the ways in which they differ from traditional PC-based systems. Outline the need for such systems and why the

PC architecture is generally incapable of meeting such high-end computing requirements for the most part.

2. Introduce the concept of clustering and outline the different types of clusters that can be created for the purpose of load balancing, redundancy, and parallel computing.

3. Introduce the concept of scalability, explaining why it is often better to use a cluster of computers with few processors rather than a single computer with many processors.

4. Introduce students to the concept of a Beowulf cluster and outline how such clusters are used to distribute processing across a number of different Linux servers. Explain how using a Message Passing Interface (MPI) software framework such as OpenMPI can be used in Beowulf clustering.

Scientific/Engineering Workstations

1. Briefly outline some of the different applications for Linux in the scientific and engineering marketplace, using the bullet points on Page 40 in the text as a guide.

Office/Personal Workstations

1. Provide students with an overview of some of the office and personal applications typically included in a Linux distribution. From a Linux GUI, demonstrate a word processing and spreadsheet application and show students the different formats in which files can be saved for compatibility purposes with Windows and other OSs. Use the bullet points on Page 41 in the text as a guide.

Cybersecurity Workstation

1. Describe what is meant by the term cybersecurity.

2. Distinguish between security vulnerabilities (called a vulnerability assessment) and a penetration test) explaining when to use each tool.

3. Emphasize the importance of continually monitoring the security of systems and networks to determine when a system has been breached, as well as performing forensic analysis to investigate the nature of the breach and the damage incurred.

4. Mention that most of the tools for performing a vulnerability assessment and penetration test, as well as the tools for detecting and investigating security breaches are exclusively for Linux systems.

5. Explain that several cybersecurity-focused Linux distributions, such as Kali Linux, ship with most of these tools preinstalled for cybersecurity use.

Teaching Tip	Students can obtain and learn more about Kali Linux at: https://www.kali.org

Mobile Devices

1. Discuss the importance of Linux-based smartphone and tablet OSs. Mention that the most notable of these was Google Android in 2008.

2. Further discuss why developer support for Android has grown exponentially in recent years.

3. Review other Linux distributions on mobile devices:
   - AsteroidOS
   - postmarketOS
   - Sailfish OS
   - SHR
   - Tizen
   - Ubuntu Touch

Quick Quiz 2

1. What Internet service provides content caching services for clients?
   a. DNS
   b. Firewall
   c. Proxy
   d. FTP
   Answer: C

2. Which e-mail service is responsible for sending messages between services?
   a. MTA
   b. MDA
   c. MUA
   d. MHA
   Answer: A

3. DNS can be used to map which of the following types of names to IP addresses?
   a. DHCP ID
   b. NetBIOS names
   c. FQDNs
   d. TCP names
   Answers: C

4. Which of the following are a collection of programs and tools designed to allow for the creation, modification, maintenance, and access of information from databases?

a. Network File Systems (NFSs)
b. Berkeley Software Distribution (BSD)
c. Multiplexed Information and Computing Service (MULTICS)
d. Database Management Systems (DBMSs)
Answer: D

5. The ability for a computer to increase workload as the number of processors increase is known as _____?
a. clustering
b. scalability
c. multitasking
d. process
Answer: B

Class Discussion Topics

1. Have students discuss their feelings on hacker culture in general. Do they feel that it has a positive or negative impact on the world of computing? If they owned their own company, would they consider hiring someone who would describe himself or herself as a hacker? Why or why not? What impact do students think the media has in the portrayal of hackers and hacker culture?

2. Ask students which of them have worked with Linux in the past and 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 to find examples of different open source, closed source, freeware, and shareware applications. Have them create a table that outlines the pros and cons of each software type.

2. Have students research online to create a list of what they consider to be the best Linux resources available, along with the types of information that the sites provide. Once completed, compile the lists into a master list to be distributed to all students.

Additional Resources

1. Linux.org: https://www.linux.org/

2. Linux Format Magazine: https://www.linuxformat.com/

3. The Linux Kernel Archives: https://www.kernel.org/

4. The Linux Journal: https://www.linuxjournal.com/

Key Terms

- **AIX** A version of UNIX developed by IBM.
- **Android** A mobile Linux-based operating system currently developed by Google's Open Handset Alliance.
- **application (app)** The software that runs on an operating system and provides the user with specific functionality (such as word processing or financial calculation).
- **Arch** A very lightweight and customizable Linux distribution. Due to its focus on simplicity and customization, it is often used within specialized environments and on small footprint systems.
- **artistic license** An open source license that allows source code to be distributed freely but changed only at the discretion of the original author.
- **asymmetric encryption** An encryption algorithm used to protect data before it is transmitted on the network. It uses a pair of keys that are uniquely generated on each computer system exchanging the data: a public key and a private key.
- **authentication** The process of logging in to a computer system using a valid username and password before gaining access to the user interface.
- **Beowulf clustering** A popular and widespread method of clustering computers together to perform useful tasks using Linux.
- **BSD (Berkeley Software Distribution)** A version of UNIX developed out of the original UNIX source code and given free to the University of California at Berkeley by AT&T.
- **certificate** A public key that has been digitally signed by a Certificate Authority (CA).
- **Certificate Authority** A trusted third-party computer by which all public keys are digitally signed.
- **closed source software** The software whose source code is not freely available from the original author; Windows 98, is an example.
- **cloud** Another term for the Internet.
- **cloud platform** A series of software components that are installed on servers distributed across the Internet and provide services to a large number of Internet users.
- **cloud provider** An organization that hosts its own private data center that is accessible to other computers across the Internet.
- **cluster** A grouping of several smaller computers that function as one large supercomputer.
- **clustering** The act of making a cluster; *see also* cluster.
- **container** A subset of an existing operating system that provides a unique service on the network in an Infrastructure as a Service (IaaS) cloud system.
- **cracker** A person who uses computer software maliciously for personal profit.
- **cybersecurity** The collectively technologies and processes used to analyze existing security and provide data protections.
- **database** An organized set of data.
- **Database Management System (DBMS)** Software that manages databases.
- **Debian** A well-known distribution of Linux.
- **developmental kernel** A Linux kernel whose minor number is odd and has been recently developed yet not thoroughly tested.

- **device driver** A piece of software containing instructions that the kernel of an operating system uses to control and interact with a specific type of computer hardware.
- **digital signature** A message that has been encrypted using a private key.
- **distribution** A complete set of operating system software, including the Linux kernel, supporting function libraries and a variety of OSS packages that can be downloaded from the Internet free of charge. These OSS packages are what differentiate the various distributions of Linux.
- **distribution kernel** A Linux distribution that uses the same Linux kernel versions that are community developed; however, they can modify those kernels in order to provide fixes and optimizations that are specific to the distribution and used for long-term support.
- **Domain Name Space (DNS**) A service that provides the proper FQDN to IP mapping, and quickly returns the requested IP address to your browser.
- **Dynamic Host Configuration Protocol (DHCP)** A client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information.
- **flavor** A term that refers to a specific type of UNIX operating system. For example, Solaris and BSD are two flavors of UNIX.
- **Free Software Foundation (FSF)** An organization, started by Richard Stallman, that promotes and encourages the collaboration of software developers worldwide to allow the free sharing of source code and software programs.
- **freeware** Software distributed by the developer at no cost to the user.
- **frequently asked questions (FAQs)** An area on a Web site where answers to commonly posed questions can be found.
- **fully qualified domain name (FQDN)** A string of words identifying a server on the Internet.
- **Gentoo** A well-known distribution of Linux.
- **GNU** An acronym that stands for “GNU’s Not UNIX.”
- **GNU General Public License (GPL)** A software license ensuring that the source code for any OSS will remain freely available to anyone who wants to examine, build on, or improve upon it.
- **GNU Network Object Model Environment (GNOME)** One of the two competing graphical user interface (GUI) environments for Linux.
- **GNU Project** A free operating system project started by Richard Stallman.
- **graphical user interface (GUI)** The component of an operating system that provides a user-friendly interface comprising graphics or icons to represent desired tasks. Users can point and click to execute a command rather than having to know and use proper command-line syntax.
- **GUI environment** A GUI core component such as X Windows, combined with a window manager and desktop environment that provides the look and feel of the GUI. Although functionality might be similar among GUI environments, users might prefer one environment to another due to its ease of use.
- **hacker** A person who explores computer science to gain knowledge. It should not be confused with the term *cracker*.
- **hardware** The tangible parts of a computer, such as the network boards, video card, hard disk drives, printers, and keyboards.

- **hardware platform** A particular configuration and grouping of computer hardware, normally centered on and determined by processor type and architecture.
- **hot fix** A solution made by a closed source vendor that fixes a software bug.
- **HOWTO** A task-specific instruction guide to performing any of a wide variety of tasks; freely available from the Linux Documentation Project at https://www.tldp.org/
- **HP-UX** A version of UNIX developed by Hewlett-Packard.
- **Infrastructure as a Service (IaaS)** A cloud service where the cloud provider provides the hardware and storage within a data center only, and companies install, manage, and access their own virtualized operating systems within that data center via cloud platform software.
- **Internet Protocol (IP) address** A unique string of numbers assigned to a computer to uniquely identify it on the Internet.
- **iOS** A mobile version of UNIX developed by Apple for use on iPhone, iPod, and iPad devices.
- **K desktop environment (KDE)** One of the two competing graphical user interfaces (GUI) available for Linux
- **Kerberos** An authentication protocol.
- **kernel** The central, core program of the operating system. The shared commonality of the kernel is what defines Linux; the differing OSS applications that can interact with the common kernel are what differentiate Linux distributions.
- **key** A random component used to modify the steps within an encryption algorithm.
- **Linus Torvalds** A Finnish graduate student who coded and created the first version of Linux and subsequently distributed it under the GNU Public License.
- **Linux** A software operating system originated by Linus Torvalds. The common core, or kernel, continues to evolve and be revised. Differing OSS bundled with the Linux kernel is what defines the wide variety of distributions now available.
- **Linux Documentation Project (LDP)** A large collection of Linux resources, information, and help files supplied free of charge and maintained by the Linux community.
- **Linux Mint** A well-known distribution of Linux.
- **Linux User Group (LUG)** The open forums of Linux users who discuss and assist each other in using and modifying the Linux operating system and the OSS run on it. There are LUGs worldwide.
- **load balancing** A router feature that allows firewalls and security appliances to separate requests for a specific resource, such as a website, across several servers that provide the same service.
- **macOS** A version of UNIX developed by Apple for use on Apple desktop computers and servers.
- **Mail Delivery Agent (MDA)** The service that downloads e-mail from a mail transfer agent.
- **Mail Transfer Agent (MTA)** An e-mail server.
- **Mail User Agent (MUA)** A program that allows e-mail to be read by a user.
- **major number** The number preceding the first dot in the number used to identify a Linux kernel version. It is used to denote a major change or modification.
- **Message Passing Interface (MPI)** A system that is used on Beowulf clusters to pass information to several separate computers in a parallel fashion.

- **MINIX** Mini-UNIX created by Andrew Tannenbaum. Instructions on how to code the kernel for this version of the UNIX operating system were publicly available. Using this as a starting point, Linus Torvalds improved this version of UNIX for the Intel platform and created the first version of Linux.
- **minor number** The number following the first dot in the number used to identify a Linux kernel version, denoting a minor modification. If odd, it is a version under development and not yet fully tested. See also *developmental kernel* and *production kernel*.
- **Multiplexed Information and Computing Service (MULTICS)** A prototype time-sharing operating system that was developed in the late-1960s by AT&T Bell Laboratories.
- **multitasking** A type of operating system that has the capability to manage multiple tasks simultaneously.
- **multiuser** A type of operating system that has the capability to provide access to multiple users simultaneously.
- **Network Time Protocol (NTP)** A protocol that provides the current time to operating systems.
- **newsgroup** An Internet protocol service accessed via an application program called a newsreader. This service allows access to postings (e-mails in a central place accessible by all newsgroup users) normally organized along specific themes. Users with questions on specific topics can post messages, which can be answered by other users.
- **Next generation Firewall (NGFW)** Security appliances that provide multiple security functions. Also known as Unified Threat .Management (UTM) appliances.
- **Open Source Software (OSS)** The programs distributed and licensed so that the source code making up the program is freely available to anyone who wants to examine, utilize, or improve upon it.
- **openSUSE** One of the most popular and prevalent distributions of Linux, originally developed in Europe.
- **operating system (OS)** The software used to control and directly interact with the computer hardware components.
- **package manager** The software used to install, maintain, and remove other software programs by storing all relevant software information in a central software database on the computer.
- **penetration test** A tool used by cybersecurity professionals to break into systems to test the strength of their security measures.
- **Platform as a Service (PaaS)** A cloud service that allows a company to create their own Web apps and services that are hosted by another cloud provider.
- **private cloud** A private data center that is accessible to other computers across the Internet.
- **private key** Part of the public key/private key pair used in asymmetric encryption. The private key is used only by the system and never distributed.
- **process** A program loaded into memory and running on the processor, performing a specific task.
- **production kernel** A Linux kernel whose minor number (the number after the dot in the version number) is even and which is, therefore, deemed stable for use after widespread testing.

- **program** A set of instructions that knows how to interact with the operating system and computer hardware to perform a specific task; stored as a file on some media (for example, a hard disk drive).
- **programming language** The syntax used for developing a program. Different programming languages use different syntaxes.
- **proxy server** A server or hardware device that requests Internet resources on behalf of other computers.
- **public key** Part of the public key/private key pair used in asymmetric encryption. The public key is freely distributed to any other host on the network.
- **Public Key Infrastructure (PKI)** An organization that installs one or more CAs.
- **Red Hat** One of the most popular and prevalent distributions of Linux in North America, distributed and supported by Red Hat Inc. Fedora is a Red Hat-based Linux distribution.
- **revision number** The number after the second dot in the version number of a Linux kernel, which identifies the certain release number of a kernel.
- **router** A computer running routing software, or a special-function hardware device providing interconnection between networks; it contains information regarding the structure of the networks and sends information from one component network to another.
- **scalability** The capability of computers to increase workload as the number of processors increases.
- **search engine** An Internet Web site, such as *www.google.com*, where you simply enter a phrase representing your search item and receive a list of Web sites that contain relevant material.
- **security appliance** Specialized server hardware running Linux that may provide routing, firewall, and proxy services alongside additional advanced security services.
- **server** A computer configured to allow other computers to connect to it from across a network.
- **server services** The services that are made available for other computers across a network.
- **shareware** The programs developed and provided at minimal cost to the end user. These programs are initially free but require payment after a period of time or a certain amount of usage.
- **software** The programs stored on a storage device in a computer that provide a certain function when executed.
- **Software as a Service (SaaS)** A cloud service that refers to hosting a service (and the associated data) within a cloud environment, where users can access the service across the Internet.
- **source code** The sets of organized instructions on how to function and perform tasks that define or constitute a program.
- **symmetric encryption** An encryption algorithm where data can be decrypted by reversing the algorithm using the same key that was used to encrypt it.
- **system service** The additional functionality provided by a program that has been incorporated into and started as part of the operating system.
- **tarball** A compressed archive of files containing scripts that install Linux software to the correct locations on a computer system.

- **total cost of ownership (TCO)** The full sum of all accumulated costs, over and above the simple purchase price of utilizing a product. Includes training, maintenance, additional hardware, and downtime.
- **Ubuntu** A major Linux distribution that is widely used in North America.
- **Unified Threat Management (UTM)** Security appliances that provide multiple security functions. Also known as Next Generation Firewall (NGFW) appliances.
- **UNIX** The first true multitasking, multiuser operating system, developed by Ken Thompson and Dennis Ritchie, from which Linux was originated.
- **user** A person who uses a computer.
- **user interface** The interface the user sees and uses to interact with the operating system and application programs.
- **vulnerability assessment** A tool used by cybersecurity professionals to scan key computers and networks for security vulnerabilities.
- **workstation** A computer used to connect to services on a server.
- **workstation services** The services that are used to access shared resources on a network server.
- **X Windows** The core component of the GUI in Linux.

Technical Notes for Hands-On Projects

Chapter 1 contains no hands-on projects and, as such, does not require any specific classroom setup. However, students will require access to the Internet to complete the additional exercises outlined in this file.

Guide to Linux+, Fifth Edition
Chapter 1 Solutions

Review Questions

1. Every computer consists of physical components and non-physical components. The non-physical components of a computer that understand how to work with the physical components are referred to as:
    a. hardware
    b. records
    c. software
    d. processors

Answer: c

2. The operating system software is necessary for a computer to function. True or False?

Answer: True

3. Linux is a _____ and _____ operating system.
    a. production, stable
    b. multiuser, multitasking
    c. processing, operating
    d. large, useful

Answer: b

4. The core component of the Linux operating system is the Linux kernel. If you were a Linux systems administrator for a company, when would you need to upgrade your Linux kernel? (Choose all that apply.)
    a. when you need support in Linux for new hardware
    b. when you need another user interface
    c. when you need to increase the stability of Linux
    d. when you need to use kernel modules

Answer: a, c

5. Which of the following kernels are developmental kernels? (Choose all that apply.)
    a. 2.3.4
    b. 3.5.5
    c. 4.1-rc5
    d. 4.4.4

Answer: a, c

6. Many types of software are available today. Which type of software does Linux represent?
    a. open source software
    b. closed source software
    c. freeware
    d. shareware

Answer: a

7. Which of the following are characteristics of Open Source Software? (Choose all that apply.)
    a. The value of the software is directly related to its price.
    b. The software is developed collaboratively.
    c. The source code for software is available for a small fee.
    d. Any bugs are fixed quickly.

Answer: b, d

8. To which license does Linux adhere?
    a. open license
    b. artistic license
    c. GNU General Public License
    d. free source license

Answer: c

9. What are some good reasons for using Linux in a corporate environment? (Choose all that apply.)
    a. Linux software is unlikely to become abandoned by its developers.
    b. Linux is secure and has a lower total cost of ownership than other operating systems.
    c. Linux is widely available for many platforms and supports many programming languages.
    d. Most Linux software is closed source.

Answer: a, b, c

10. Which of the following are common methods for gaining support for Linux?
    a. HOWTO documents at *www.tldp.org*
    b. a local Linux User Group
    c. Internet newsgroups
    d. all the above

Answer: d

11. Which two people are accredited with creating the UNIX operating system? (Choose two answers.)
    a. Dennis Ritchie
    b. Richard Stallman
    c. Linus Torvalds
    d. Ken Thompson

Answer: a, d

12. Who formed the Free Software Foundation to promote open development?
    a. Dennis Ritchie
    b. Richard Stallman
    c. Linus Torvalds
    d. Ken Thompson

Answer: b

13. Which culture embraced the term “GNU” (GNU’s Not UNIX) and laid the free software groundwork for Linux?
    a. the hacker culture
    b. the MIT culture
    c. the cracker culture
    d. the Artificial Intelligence culture

Answer: a

14. Linux was developed by _____ to resemble the _____ operating system.
    a. Linus Torvalds, MINIX
    b. Linux Torvalds, GNU
    c. Richard Stallman, GNU
    d. Richard Stallman, MINIX

Answer: a

15. When the core components of the Linux operating system are packaged together with other Open Source Software, it is called a:

a. new kernel
b. new platform
c. Linux distribution
d. GNU Project

Answer: c

16. Which common GUI environments are available in most Linux distributions? (Choose all that apply.)
a. GNOME
b. CDE
c. KDE
d. RPM

Answer: a, c

17. Which of the following are factors that determine which Linux distribution a user will use? (Choose all that apply.)
a. package manager support
b. hardware platform
c. kernel features
d. language support

Answer: a, b, c

18. What is the most common open source Web server available for Linux?
a. Samba
b. Apache
c. Squid
d. OpenStack

Answer: b

19. Which of the following can be used on Linux to provide file and print services?
a. Samba
b. Apache
c. Squid
d. OpenStack

Answer: a

20. Which of the following Linux distributions is likely to be used by a CyberSecurity worker?
a. Fedora
b. Ubuntu
c. Kali
d. Gentoo

Answer: c

Hands-on Projects

[There are no Hands-On Projects for Chapter 1.]

Discovery Exercises

Discovery Exercise 1

Answers will vary. Common considerations include total cost of ownership, whether Oracle can run on Linux, and whether hardware will be newly purchased or the Windows servers upgraded. The distribution

to be chosen will likely be a major distribution, such as Red Hat, openSUSE, Ubuntu, or Debian. A report on Linux advantages could include the following points:

- Linux and Linux software are continuously developed and rarely become abandoned projects.
- Bugs are fixed quickly in Open Source Software.
- Hardware support is abundant.
- Linux support is abundant.
- Customizing Linux is easy.
- The custom UNIX software used in the company can port to Linux easily.
- Linux and most software needed for this task are freely obtainable.

Discovery Exercise 2

Answers will vary. The GPL stands for the GNU General Public License and allows Open Source Software such as Linux to be freely available and improved for an indefinite period of time. The Open Source business model is different from conventional software; the value of software is based on its use and its use generates profit indirectly by means of support, hardware sales, and so on. Open Source Software is scrutinized by several developers worldwide via the Internet; thus, software bugs and security loopholes are found and fixed quickly. This makes Open Source Software very high-quality software compared to commercial software. Websites that can be used for more information include but are not limited to the following list:

www.catb.org/~esr/writings/cathedral-bazaar/
www.gnu.org
www.opensource.org

Discovery Exercise 3

Answers will vary. Benefits to publishing software under the GPL include rapid collaborative development in new areas, regular feedback, greater usage, and low chance of obsolescence. Companies can benefit from the free development of other open source developers and reduce the cost of creating software packages by contributing people to only those open source projects of corporate interest. To release software as open source, you must read the open source definition and either subscribe to a license such as the GPL or create a new one that is accepted at *www.opensource.org*. Then, you must coordinate the development by providing an Internet-accessible means of communication between developers such as newgroups or *www.sourceforge.net*. Both aforementioned websites also list the rules and resources available to those who want to gain more knowledge about the open source procedure.

Discovery Exercise 4

Not available.

Discovery Exercise 5

Not available.

Discovery Exercise 6

Answers will vary. The philosophy of the hacker culture dictates that the sharing of knowledge advances computer science and enhances personal development. Software is distributed for free along with the source code; should one person change that source code, that person must redistribute that source code so that others can benefit from the changes or give appropriate feedback (for example, fix bugs). The hacker culture is a global culture tied together by Internet communication (emails, newgroups, FTP, and so on) and

produces high-quality software in short periods of time. The software produced by the hacker culture is developed to meet a certain demand and, thus, is rarely abandoned. The bulleted list required for this Discovery Exercise should mention the previous topics but can also mention any other aspect of Linux, GNU, FSF, OSS, or UNIX. The anticipated questions will also vary but could include questions regarding the profit structure of OSS, personal benefits to developing OSS, and school-related benefits of developing OSS.

Discovery Exercise 7

Not available.