NETWORKING

Guide to Operating Systems

Fifth Edition

Greg Tomsho

Guide to Operating Systems

Fifth Edition

Greg Tomsho

CENGAGE Learning®

Australia • Brazil • Mexico • Singapore • United Kingdom • United States

CENGAGE Learning

Guide to Operating Systems, 5th edition

Greg Tomsho

GM, Science, Technology, & Math: Balraj Kalsi

Senior Product Director, Computing: Kathleen McMahon

Product Team Manager: Kristin McNary

Associate Product Manager: Amy Savino

Director, Development: Julia Caballero

Content Development Manager, Computing: Leigh Hefferon

Managing Content Developer: Emma Newsom

Senior Content Developer: Natalie Pashoukos

Product Assistant: Abigail Pufpaff

Vice President, Marketing Services: Jennifer Ann Baker

Marketing Coordinator: Cassie Cloutier

Senior Content Project Manager: Brooke Greenhouse

Production Director: Patty Stephan

Art Director: Diana Graham

Cover Image(s): © iStock.com/xiaoke ma

Library of Congress Control Number: 2016942762

ISBN: 978-1-305-10764-9

Cengage Learning
20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with employees residing in nearly 40 different countries and sales in more than 125 countries around the world. Find your local representative at **www.cengage.com**.

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

To learn more about Cengage Learning, visit **www.cengage.com**.

Purchase any of our products at your local college store or at our preferred online store **www.cengagebrain.com**.

Printed in the United States of America
Print Number: 01 Print Year: 2016

Brief Contents

Table of Contents

Introduction

If you use a computer, you also use a computer operating system to tap into the computer's power. The more you know about a computer's operating system, the more you are able to enjoy the full versatility of your computer. This book opens the door to understanding your computer's operating system. Also, the book enables you to understand many types of operating systems so you can compare the advantages of each for your personal and professional use.

In this book, you learn about the most popular operating systems in use today:

- Windows, with emphasis on Windows 10 and Windows Server 2012, and coverage of Windows Server 2016
- UNIX/Linux, with emphasis on Fedora 23
- Mac OS X, with emphasis on El Capitan

The book starts at a basic level and builds with each chapter to put you on track to become an accomplished user of each operating system.

You learn the operating systems in clear language through a hands-on, practical approach. An advantage of studying several operating systems is that you can compare the functions of each side-by-side as you learn. If you are taking an introductory operating systems course or an operating systems survey course, this book offers a strong foundation for mastering operating systems. Also, if you are preparing for one or more computer certifications, such as for hardware systems, networking, programming, or security, you'll find this book provides a vital background for your preparations. The book is particularly useful as background for the CompTIA A+ certification. If you are relatively new to computers, the book starts with the basics to build your confidence. If you are more experienced in computers, you'll find lots of useful information to further build your repertoire of knowledge and experience.

The Intended Audience

Guide to Operating Systems, Fifth Edition is written in straightforward language for anyone who uses a computer and wants to learn more. No prior computer experience is required, although some previous basic experience with a computer is helpful. The hands-on projects in this book use a variety of operating systems. You can learn the concepts if you have access to one or a combination of the operating systems presented. The more operating systems that are available to you, the better the opportunity to compare their features. For the most part, the projects can be performed in a classroom, computer lab, or at home.

What's New to this Edition

Guide to Operating Systems, Fifth Edition is extensively updated to include the most current operating systems and operating system features. This includes all-new coverage of Windows 10, Windows Server 2012, Windows Server 2016, Fedora Linux with the GNOME desktop, and Mac OS X El Capitan. Coverage of legacy operating systems is greatly reduced to provide mainly an historical perspective.

The book also includes extensive updates for new hardware and new operating system installation and management activities. New hardware coverage includes the latest CPUs and peripheral devices. The interaction of operating systems and new storage devices is also significantly updated, as well as new networking capabilities, including wireless networking advances and cloud computing. An entire chapter has been added for operating system virtualization, including VMware, Microsoft Hyper-V, and VirtualBox.

Screen captures, figures, and tables are virtually all new. The hands-on projects are fully updated or are all new for the new operating systems. A new end-of-chapter section called Challenge Labs gives readers one or more hands-on activities that require research and synthesis of information already learned.

Chapter Descriptions

The chapter coverage is balanced to give you a full range of information about each topic. The following is a summary of what you will learn in each chapter. Besides the instruction provided throughout the chapter text, you can build on your knowledge and review your progress using the extensive hands-on projects, challenge labs, case projects, key terms, and review questions at the end of each chapter.

- *Chapter 1: Operating Systems Fundamentals* gives you a basic introduction to operating systems, including the types of operating systems and how they work. You also learn about the history of operating systems.

- *Chapter 2: Popular Operating Systems* presents in-depth descriptions of modern Windows OSs, Linux and UNIX, and Mac OS X. You also briefly learn about earlier Microsoft operating systems. This chapter gives you a starting point from which to compare features of operating systems and to understand advancements in the latest versions.
- *Chapter 3: The Central Processing Unit (CPU)* enables you to understand how processors work and the essential characteristics of modern processors. The chapter concludes with an overview of popular modern processors.
- *Chapter 4: File Systems* explains the functions common to all file systems and then describes the specific file systems used by different operating systems, from a brief introduction to FAT to more in-depth coverage of NTFS, ufs/ext, HFS, and HFS+.
- *Chapter 5: Installing and Upgrading Operating Systems* shows you how to prepare for installing operating systems and then shows you how to install each operating system discussed in this book. You learn about installing operating systems from scratch and how to upgrade operating systems.
- *Chapter 6: Configuring Input and Output Devices* explains how devices such as monitors, keyboards, mice, disk drives, network cards, and other devices interface with operating systems. You learn about the latest input and output technologies for modern operating systems and computers.
- *Chapter 7: Using and Configuring Storage Devices* describes popular storage devices, including hard drives, removable drives, RAID, CD and DVD technologies, flash and solid-state drive storage, network storage, USB devices, and the latest emerging technologies. Storage device configuration is covered for the operating systems and you learn how to perform backups for Windows, UNIX/Linux, and Mac OS X operating systems.
- *Chapter 8: Virtualization and Cloud Computing Fundamentals* introduces you to virtualization, its terminology, and some of the popular virtualization products that can be used on the OSs this book discusses.
- *Chapter 9: Configuring a Network Connection* provides an introduction to how networks function, including network technologies and protocols. You learn how to configure protocols in each operating system and you learn about the basic structure of local and wide area networks. You also learn how operating systems interface with networks.
- *Chapter 10: Sharing Resources and Working with Accounts* shows you many ways to share resources through a network, including sharing disks, folders, and printers. Besides covering how to share resources, the chapter also discusses how to secure them through accounts, groups, and access privileges.
- *Chapter 11: Operating Systems Management and Maintenance* presents many techniques for maintaining systems, such as cleaning up unused files, defragmenting disks, making file system repairs, tuning virtual memory, and addressing problems. The chapter also addresses planning for backups and how to tune systems for top performance.
- *Appendix A: Operating System Command-Line Commands* shows you how to access the command line in each operating system and presents tables that summarize general and network commands. This appendix provides a place to quickly find or review the operating system commands.

Features

To aid you in fully understanding operating system concepts, there are many features in this book designed to improve its pedagogical value.

- **Chapter Objectives.** Each chapter in this book begins with a detailed list of the concepts to be mastered within that chapter. This list provides you with a quick reference to the contents of each chapter as well as a useful study aid.
- **Illustrations and Tables.** Numerous illustrations of operating system screens and concepts aid you in the visualization of common setup steps, theories, and concepts. In addition, many tables provide details and comparisons of both practical and theoretical information.
- ***From the Trenches* Stories and Examples.** Each chapter contains boxed text with examples from the author's extensive experience to add color through real-life situations.
- **Chapter Summaries.** Each chapter's text is followed by a summary of the concepts it has introduced. These summaries provide a helpful way to recap and revisit the ideas covered in each chapter.
- **Key Terms.** A listing of the terms that were introduced throughout the chapter, along with definitions, is presented at the end of each chapter.
- **Review Questions.** The end-of-chapter assessment begins with a set of review questions that reinforce the ideas introduced in each chapter.
- **Hands-On Projects.** The goal of this book is to provide you with the practical knowledge and skills to install and administer desktop and server operating systems as they are employed for personal and business use. To this end, along with theoretical explanations, each chapter provides numerous hands-on projects aimed at providing you with real-world implementation experience.
- **Critical Thinking Sections.** The end-of-chapter Critical Thinking section gives you more opportunities for hands-on practice with **Challenge Labs**, which enable you to use the knowledge you've gained from reading the chapter and performing hands-on projects to solve more complex problems without step-by-step instructions. This section also includes **Case Projects** that ask you to evaluate a scenario and decide on a course of action to propose a solution. These valuable tools help you sharpen decision-making, critical thinking, and troubleshooting skills.

Text and Graphic Conventions

Whenever appropriate, additional information and activities have been added to this book to help you better understand what is being discussed in the chapters. Icons throughout the text alert you to additional materials. The icons used in this textbook are as follows:

The Note icon is used to present additional helpful material related to the subject being described.

Tips are included from the author's experience to provide extra information about how to configure an operating system, apply a concept, or solve a problem.

Cautions are provided to help you anticipate potential problems or mistakes so you can prevent them from happening.

Each hands-on project in this book is preceded by the Hands-On Projects icon and a description of the practical exercise that follows.

Case Project icons mark each case project. Case projects are more involved, scenario-based assignments. In each extensive case example, you are asked to implement what you have learned.

MindTap

MindTap for *Guide to Operating Systems* is an online learning solution designed to help students master the skills they need in today's workforce. Research shows that employers need critical thinkers, troubleshooters, and creative problem-solvers to stay relevant in our fast-paced, technology-driven world. MindTap helps users achieve this with assignments and activities that provide hands-on practice, real-life relevance, and mastery of difficult concepts. Students are guided through assignments that progress from basic knowledge and understanding to more challenging problems.

All MindTap activities and assignments are tied to learning objectives. The hands-on exercises provide real-life application and practice. Readings and "Whiteboard Shorts" support the lecture, while "In the News" assignments encourage students to stay current. Pre- and post-course assessments allow you to measure how much students have learned using analytics and reporting that makes it easy to see where the class stands in terms of progress, engagement, and completion rates. You can use the existing content and learning path or pick and choose how the material will wrap around your own content. You control what the students see and when they see it. Learn more at *www.cengage.com/mindtap/*.

Instructor Resources

Everything you need for your course in one place! This collection of book-specific lecture and class tools is available online via *www.cengage.com/login*. Access and download PowerPoint presentations, images, the Instructor's Manual, and more.

- *Electronic Instructor's Manual*—The Instructor's Manual that accompanies this book includes additional instructional material to assist in class preparation, including suggestions for classroom activities, discussion topics, and additional quiz questions.

- *Solutions*—The instructor's resources include solutions to all end-of-chapter material, including review questions and case projects.
- *Cengage Learning Testing Powered by Cognero*—This flexible, online system allows you to do the following:
    - Author, edit, and manage test bank content from multiple Cengage Learning solutions.
    - Create multiple test versions in an instant.
    - Deliver tests from your LMS, your classroom, or wherever you want.
- *PowerPoint presentations*—This book comes with Microsoft PowerPoint slides for each chapter. They're included as a teaching aid for classroom presentations, to make available to students on the network for chapter review, or to be printed for classroom distribution. Instructors, please feel free to add your own slides for additional topics you introduce to the class.
- *Figure files*—All the figures and tables in the book are reproduced in bitmap format. Similar to the PowerPoint presentations, they're included as a teaching aid for classroom presentations, to make available to students for review, or to be printed for classroom distribution.

About the Author

Greg Tomsho is director of the Computer Networking Technology Department and Cisco Academy at Yavapai College in Prescott, Arizona. He has earned the CCNA, MCTS, MCSA, A+, Network+, Security+, Server+, and Linux+ certifications. A former software engineer, technical support manager, and IT director, he has more than 30 years of computer and networking experience. His other books include *MCSA Guide to Installing and Configuring Microsoft Windows Server 2012/R2*; *MCSA Guide to Administering Microsoft Windows Server 2012/R2*; *MCSA Guide to Configuring Advanced Microsoft Windows Server 2012/R2 Services*; *MCTS Guide to Windows Server 2008 Active Directory Configuration*; *MCTS Guide to Microsoft Windows Server 2008 Applications Infrastructure Configuration*; *Guide to Networking Essentials*; *Guide to Network Support and Troubleshooting*; and *A+ CoursePrep ExamGuide*.

Acknowledgments

I would like to thank the team at Cengage Learning for this opportunity to improve and expand on the fifth edition of this book. This team includes Kristin McNary, Product Team Manager; Natalie Pashoukos, Senior Content Developer; Brooke Greenhouse, Senior Content Project Manager; and Serge Palladino, Nicole Spoto, and Danielle Shaw of Manuscript Quality Assurance, who tested projects for accuracy. Thanks also go to my development editor, Dan Seiter, for his guidance in creating a polished product. Additional praise and special thanks go to my beautiful wife, Julie; our daughters, Camille and Sophia; and our son, Michael. As always, they have been patient and supportive throughout the process and I truly appreciate their support.

Reviewers

Guy Garrett, M.S., M.B.A.
Associate Professor, Cybersecurity & Information Technology
Manager, Network Systems Technology/Cybersecurity Program
Gulf Coast State College
Panama City, FL

Emily Harrington
CIT Coordinator/Faculty
Pitt Community College
Winterville, NC

Todd Koonts, MSIT, CCE
Program Chair
CTI/Information Assurance and Digital Forensics
Central Piedmont Community College
Charlotte, NC

Before You Begin

The importance of a solid lab environment can't be overstated. This book contains hands-on projects that require a variety of operating systems, including Windows 10, Linux Fedora 23, and Mac OS X El Capitan. Using virtualization can simplify the lab environment. For example, you can use VMware Player, VMware Workstation, VirtualBox, and other products to install Windows and Linux in a virtual machine, regardless of the OS running on your physical computer. Installing Mac OS X in a virtual machine running on Windows requires some creativity, but it can be done. If you want to run El Capitan as a VMware virtual machine running on Windows, do a little Internet research on the topic. The following section lists the requirements for completing hands-on activities and challenge labs.

Lab Setup Guide

Most of the hands-on projects and challenge labs require a Windows 10, Linux Fedora 23, or Mac OS X El Capitan computer. The computers should have a connection to the Internet, but only a few of the activities actually require Internet access. The use of virtual machines is highly recommended.

Windows 10 Computers

- Windows 10 Enterprise or Education Edition is recommended, but other versions are acceptable
- An account that is a member of the local Administrators group
- Workgroup name: Using the default workgroup name ("Workgroup") is acceptable, but the name is not important
- Memory: 1 GB required, 2 GB or more recommended

- Hard disk 1: 60 GB or more (Windows installed on this drive)
- Hard disk 2: Unallocated 60 GB or more
- IP address via DHCP server or static if required on your network
- Internet access

Fedora 23 Computers

- Fedora 23 Linux locally installed (a live CD boot will work for some activities, but not all)
- An administrator account and access to the root password
- Memory: 1 GB
- Hard disk 1: 60 GB or more (Fedora 23 installed on this drive)
- Hard disk 2: 20 GB or more
- IP address via DHCP server or static if required on your network
- Internet access

Mac OS X Computers

- Mac OS X El Capitan
- An administrator account and access to the root password
- Memory: 1 GB
- Hard disk 1: 60 GB or more (Mac OS X El Capitan installed on this drive)
- IP address via DHCP server or static if required on your network
- Internet access

Additional Items

- Windows 10 installation media (DVD or .iso file)—Using an evaluation copy is acceptable. You can download evaluation copies of Windows from *www.microsoft.com/en-us/evalcenter/*
- Windows Server 2012 R2 installation media (DVD or .iso file)—Using an evaluation copy is acceptable
- Linux Fedora 23 installation media (DVD or .iso file)

chapter 1

Operating Systems Fundamentals

After reading this chapter and completing the exercises, you will be able to:

- Explain basic operating system concepts
- Understand the history of operating system development
- Discuss how operating systems work
- Describe the types of operating systems
- Discuss single tasking versus multitasking
- Differentiate between single-user and multiuser operating systems
- List and briefly describe current operating systems

Computers come in many and varied physical forms. There are supercomputers that perform complex computing tasks at incredible speeds, business servers that provide enterprise-level networked applications, desktop and laptop PCs, tablets, smartphones, and wearable computers. Plus, there are many devices you may not even think of as having a computer, such as those embedded in everyday devices like cars, televisions, and even dishwashers. Without an operating system, however, these devices are only a collection of electronic parts. It takes an operating system to turn a computer into a functioning device for work or play. The operating system is the software that starts the basic functions of a computer, displays documents on the computer's monitor, accesses the Internet, and runs applications—it transforms the computer into a powerful tool. There are many kinds of operating systems, but only a few have captured a wide audience. Server operating systems like Windows Server 2016 and UNIX run on network servers, and client operating systems like Windows 10, Mac OS X, and Ubuntu Linux run on desktop computers. Some operating systems are very specialized and rarely seen, such as those that run the electronics in a car. Others are ubiquitous, such as Android and iOS, which run mobile devices.

This book is your guide to the most popular operating systems. In the beginning chapters, you take an in-depth look at popular desktop or client operating systems: Windows 10, Windows 8.1, Windows 7, UNIX/Linux (particularly Linux), and Mac OS X El Capitan. Later in the book, you examine popular server operating systems: Windows Server 2016, Windows Server 2012, and UNIX/Linux. (Note that several distributions of Linux/UNIX can be either client or server operating systems.) This chapter sets the foundation for understanding desktop and server operating systems by introducing you to concepts that apply to most operating systems. With this knowledge under your belt, you will have a solid frame of reference to understand operating system specifics as they are discussed in later chapters.

About the Hands-On Projects

Be sure to read and complete the activities in the "Before You Begin" section of the Introduction. The Hands-On Projects in this book require that you first set up your lab environment so it is ready to go. The "Before You Begin" section gives you step-by-step instructions for the suggested lab configuration to use with all activities in this book.

Completing the Hands-On Projects is important because they contain information about operating systems that is best understood through hands-on experience. If for some reason you can't do some of the projects, you should at least read through each one to make sure you don't miss important information.

An Introduction to Operating Systems

Before we discuss how an operating system works, let's review the basic functions of any computer. A computer's functions and features can be broken down into the three basic tasks all computers perform: input, processing, and output. Information is input to a computer from a device such as a keyboard or from a storage device such as a hard drive; the central processing unit (CPU) processes the information, and then output is usually created. The following example illustrates the process:

- *Input*—A user running a word-processing program types the letter *A* on the keyboard, which results in sending a code to the computer representing the letter *A*.
- *Processing*—The computer's CPU determines what letter was typed by looking up the keyboard code in a table.
- *Output*—The CPU sends instructions to the graphics cards to display the letter *A*, which is then sent to the computer monitor.

The three functions described above involve some type of computer **hardware**, but the hardware is controlled and coordinated by the operating system. Without an operating system, every application you use would have to know the details of how to work with each of the hardware devices. Without the operating system to coordinate things, only one application could run at a time. So, you couldn't open a Web browser while working on a Word document, for example. The operating system can be seen as the go-between for the applications you run and the computer hardware.

In a nutshell, an **operating system** (**OS**) is a specialized computer program that provides the following features:

- *User interface*—The **user interface** provides a method for users to interact with the computer, usually with a keyboard and mouse or touch screen. A user clicks, touches, or types; the computer processes the input and provides some type of output.
- *File system*—The **file system** is the method by which an OS stores and organizes files and manages access to files on a storage device, such as a hard drive.
- *Processes and services*—A **process** is a program that's loaded into memory and run by the CPU. It can be an application a user interacts with, such as a word-processing program or a Web browser, or a program with no user interface that communicates with and provides services to other processes. This type of process is usually called a **service** in Windows and a *daemon* in Linux, and is said to run in the background because there's no user interface.
- *Kernel*—The **kernel** is the heart of the OS and runs with the highest priority. It schedules processes to run, making sure high-priority processes are taken care of first; manages memory to ensure that two applications don't attempt to use the same memory space; and makes sure I/O devices are accessed by only one process at a time, in addition to other tasks.

Each of the above OS components are discussed in more detail throughout this book. For now, let's look more closely at various types of OSs.

Desktop Versus Server Operating System

A **computer program** is a series of instructions executed by the computer's CPU. A computer program can be large and complex, like an operating system, or it can be small and fairly simple, such as a basic app running on a mobile device. What's special about an operating system program compared to an app is that the operating system is loaded when the computer is turned on and remains running until you turn the computer off. Its job is to make the computer useful so you can run apps, access the Internet, and communicate with other computers.

While there are many types of OSs, and they are designed for different purposes, this book focuses on desktop or client OSs, and server OSs. A **desktop operating system**, or *client operating system*, is typically installed on a personal computer (PC) that is used by one person at a time, and is almost always connected to a network, either wired or wirelessly. The hardware used with a client OS can be in several forms, such as:

- A full desktop computer consisting of separate components for the monitor, CPU box, keyboard, and mouse
- A portable or laptop unit that combines the monitor, CPU box, keyboard, and pointing device in an all-in-one device that is easy to carry
- A combination such as the iMac computer in which the monitor and CPU are in one unit with a separate keyboard and mouse
- A fourth category, often referred to as a 2-in-1, consists of a large tablet computer such as the iPad Pro or Microsoft Surface, along with a detachable keyboard

A **server operating system** is usually installed on a more powerful computer that typically has a wired connection to a network, and can act in many roles to enable multiple users to access information, such as e-mail, files, and software. The server hardware can also take different forms, including traditional server hardware, rack-mounted server hardware, and blade servers.

The traditional server, often used by small or medium-sized businesses, consists of a monitor, CPU box, keyboard, and mouse. **Rack-mounted servers** are CPU boxes mounted in racks that can hold multiple servers. Each rack-mounted server typically has its own power cord and network connection—but these servers often share one monitor and pointing device. Depending on the height of the rack and the height of the servers, one rack can hold a few servers or several dozen. **Blade servers** conserve even more space than rack-mounted servers; each blade server typically looks like a card that fits into a blade enclosure. The **blade enclosure** is a large box with a backplane that contains slots for blade servers; the box provides cooling fans, electrical power, connection to a shared monitor and pointing device, and even network connectivity, depending on the blade enclosure. A single blade enclosure can house over 100 blade servers. Medium-sized and large organizations use rack-mounted and blade servers to help conserve space and to consolidate server management.

Visit *www.hpe.com*, *www.dell.com*, or *www.supermicro.com* to view examples of traditional, rack-mounted, and blade servers. Also, note that the actual hardware design of rack-mounted and blade servers varies by manufacturer.

Modern desktop and server operating systems are designed to enable network communications so that the operating systems can communicate with one another over a network cable, through wireless communications, and through the Internet. Network communications enable sharing files, sharing printers, and sending e-mail.

Input and Output

One of the most basic tasks of an operating system is to take care of **input/output** (**I/O**) functions, which let other programs communicate with the computer hardware. The I/O functions take requests from the software the user runs (the application software) and translate them into low-level requests that the hardware can understand and carry out. In general, an operating system serves as an interface between application software and hardware, as shown in Figure 1-1. Operating systems perform the following I/O tasks:

- Handle input from the keyboard, mouse, and other input devices
- Handle output to the monitor and printer
- Manage network communications, such as for a local network and the Internet
- Control input/output for devices such as network interface cards
- Control information storage and retrieval using various types of storage media such as hard drives, flash drives, and DVDs
- Enable multimedia use for voice and video composition or reproduction, such as recording video from a camera or playing music through speakers

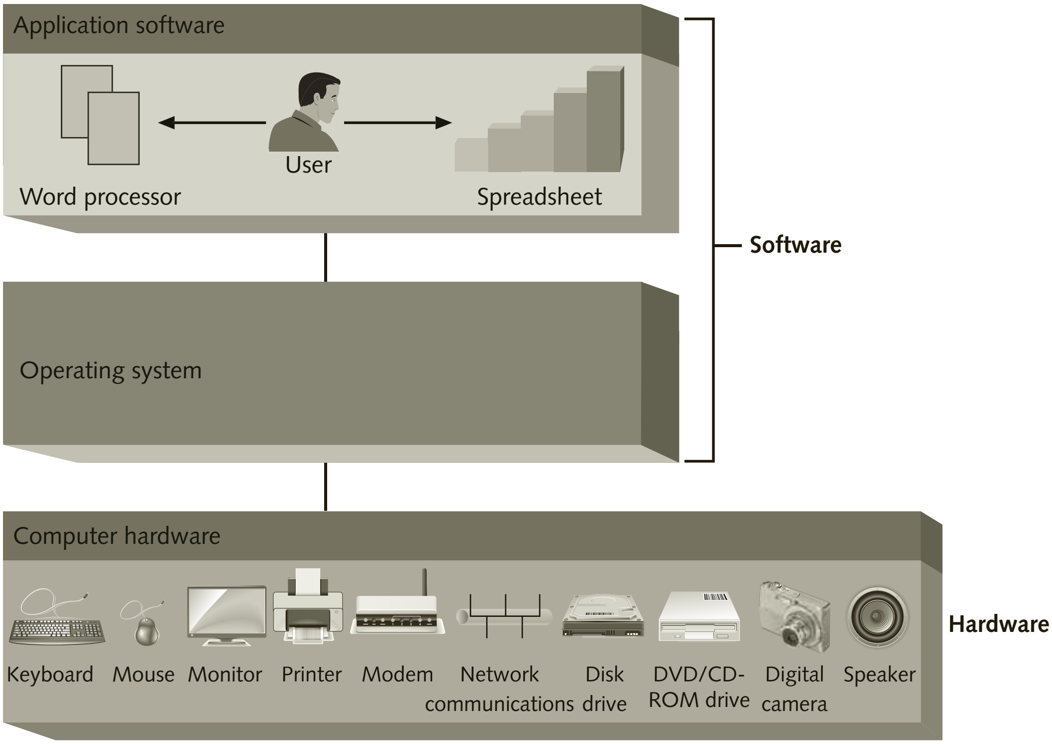

Figure 1-1 General tasks for all operating systems

A Short History of Operating Systems

The history of operating systems is a very elaborate subject. As a matter of fact, there are many books on this subject. This short history is not meant to be comprehensive; it merely presents enough background information to show how some of the features in modern PCs and PC operating systems developed.

Initially, computers were used as large automated calculators to solve all sorts of mathematical and statistical problems. Computers were extremely large, often taking up entire rooms. Although you can legitimately trace the history of today's digital computers back 100 years or more, no practical designs were used by significant numbers of people until the late 1950s. Scientists programmed these computers to perform precise tasks. The operating systems were rudimentary, often not able to do more than read punch cards or tape and write output to Teletype machines (machines resembling typewriters). A tape or deck of cards was loaded, a button was pushed on the machine to indicate the input was ready, and the machine started to read the tape and perform the operations requested. If all went well, the work was done and the output was generated. This output would be sent to the Teletype, and that was that.

Yes, there was computer history before this point, but it did not involve any sort of operating system. Any program that the computer ran had to include all logic to control the computer. Because this logic was rather complex, and not all scientists were computer scientists, the operating system was a tool that allowed non-computer scientists to use computers. The OS reduced programming work and increased efficiency. Obviously, there was not all that much to "operate" on—mainly the punch card and punch tape readers for input and the Teletype printer for output. There also was not that much to operate with; memory capacity was very limited and the processing speed of the computer was slow by our standards (but fast for that time). The art in operating systems design, therefore, largely was to keep them very small and efficient.

It took only a few decades for computer applications to evolve to appeal to a broader audience. Although computers of the late sixties and early seventies were crude by today's standards, they were quite capable and handled extremely complex tasks. These computers contributed to the development of space travel, submarine-based ballistic missiles, and a growing global financial community. Computers of this time used only a few kilobytes of RAM and rudimentary storage of only a few megabytes. This period also saw the beginning of a global, computer-based communications system called the *Internet*. Applications became logically more complex, requiring larger programs and large amounts of data.

From the Trenches ...

In the 1990s, student registration, accounting, student aid, and all other administrative functions in a state's community college system were performed on one large computer at each community college—that had only 4 MB of RAM. The system administrators of those computers considered these machines to have more than enough memory to run all administrative functions for a single college. Today, those functions are performed at each location on servers; each server is much smaller in physical size, and each uses tens of GB or more of RAM.

As always, necessity was the mother of invention. Input and output devices were created, and computer memory capacity and speed increased. With more devices to manage, operating systems became more complex and extensive, but the rule of thumb—small and fast—was still extremely important. This round of evolution, which really began to take off in the mid-seventies, included the display terminal, a Teletype machine with a keyboard that did not print on paper, but projected letters on a screen (commonly referred to as a *cathode ray tube* or *CRT*). The initial CRT was later followed by a terminal that could also show simple graphics; the terminal looked like an early computer, but it was only a monitor and a keyboard without a CPU or processing capability. The magnetic tape drive, used to store and retrieve data and programs on tape, could store more and was less operator intensive than paper tape. It was quickly followed by numerous manifestations of magnetic disks.

The next evolution was the ability to share computer resources among various programs. If a computer was very fast and could quickly switch among various programs, you could do several tasks seemingly at once and serve many people simultaneously. Some of the operating systems that evolved in this era are long lost to all but those who worked directly with them. However, there are some notable players that were responsible for setting the stage for the full-featured functionality we take for granted today. Digital Equipment Corporation's (DEC's) PDP series computers, for example, ran the DEC operating system, simply known as OS, in one version or another. A popular one was OS/8, which came in various versions, such as Release 3Q; OS/8 was released in 1968. PDP-8 computers were general-purpose machines that at one time were the top-selling computers across the world. The PDP series could also run Multics, which was the basis for the development of the first version of UNIX, a multiuser, multitasking operating system. (Multics is widely considered to be the first multiuser, multitasking operating system. You'll learn about multitasking later in the chapter.)

To find out more about the once popular PDP-8 computers, visit *www.cs.uiowa.edu/~jones/pdp8*.

The original UNIX was developed at AT&T Bell Labs in 1969 by Kenneth Thompson and Dennis Ritchie as an improvement on Multics. Later, DEC VAX computers used Virtual Memory System (VMS), a powerful, multitasking, multiuser operating system that was strong on networking. IBM mainframes made a series of operating systems popular, starting with GM-NAA I/O in the early sixties and later with System/360. Many others would follow, including CICS, which is still in use today.

Programming computers at this time was still a very complicated process best left to scientists. In the mid-1960s, right after the first interactive computer game was invented at the Massachusetts Institute of Technology (MIT), a simple programming language was developed at Dartmouth College, aimed at the nonprogrammer. It was dubbed **BASIC**, or **Beginner's All-purpose Symbolic Instruction Code**, and became a widely used programming language for many years to follow. A few years later, in 1975, Bill Gates discovered BASIC, and became interested enough to write a compiler for it. (A compiler is software that turns computer code written by people into code that is understood by computers.) Gates then sold the compiler to a company called Micro Instrumentation Telemetry Systems (MITS). MITS was the first company to produce a desktop computer that was widely accepted and could conduct useful work at the hands of any knowledgeable programmer. That same year, Gates dropped out of Harvard to dedicate his time to writing software. Other programming languages introduced around this time included Pascal, C, and other versions of BASIC supplied by various computer manufacturers. Only a couple of years later, Gates' new company (Microsoft) and others adapted popular mainframe and minicomputer programming languages, such as FORTRAN and COBOL, so they could be used on desktop computers. There were also proprietary languages that gained some popularity—languages primarily designed for database programming, for example—but they didn't last and aren't significant to this book.

The introduction of the microcomputer in the mid-1970s was probably the most exciting thing to happen to operating systems. These machines typically had many of the old restrictions, including slow speed and little memory. Many microcomputers came with a small operating system and Read-Only Memory (ROM) that did no more than provide an elementary screen, keyboard, printer, and disk input and output. Gates saw an opportunity and put together a team at Microsoft to adapt a fledging version of a new microcomputer operating system called *86-DOS*, which ran on a prototype of a new microcomputer being developed by IBM called the *personal computer*. 86-DOS was originally written by Tim Paterson (from Seattle Computer Products) as the Quick and Dirty Operating System (QDOS) for the new 8086 microprocessor. 86-DOS (or QDOS) evolved in 1980 through a cooperative effort between Paterson and Microsoft into the **Microsoft Disk Operating System**, or **MS-DOS**. MS-DOS was designed as a **command-line interface**, which means that users typed in commands instead of using the **graphical user interface (GUI)** point-and-click method that is common today.

The original MS-DOS did not offer a GUI desktop from which to click menus and icons. The command-line interface is available in modern Windows operating systems, as well as in Linux and Mac OS X. Some server administrators prefer to use a command-line interface because it offers more individualized and specialized control over the operating system. You'll have an opportunity to use command-line interfaces in Windows, Linux, and Mac OS X throughout the projects in this book.

MS-DOS became a runaway success for Microsoft, and it was the first widely distributed operating system for microcomputers that had to be loaded from disk or tape. There were earlier systems, including Control Program/Monitor (CP/M), that used some features and concepts of the existing UNIX operating system designs, but when IBM adopted MS-DOS for its PC (calling it PC DOS), the die was cast.

What did MS-DOS do? It provided the basic operating system functions described earlier in this chapter, and it was amazingly similar to what was used before on larger computers. It supported basic functions, such as keyboard, disk, and printer I/O—and communications. As time went on, more and more support functions were added, including support for such things as hard disks. Then along came the Apple Macintosh in 1984, with its GUI and mouse pointing device, which allowed users to interact with the operating system on a graphical screen. The mouse allowed users to point at or click icons or to select items from menus to accomplish tasks. Initially, Microsoft chose to wait on development of a GUI, but after Microsoft saw the successful reception of the interface on Apple computers, it developed one of its own.

When the Macintosh was introduced, it seemed light years ahead of the IBM PC. Its operating system came with a standard GUI at a time when MS-DOS was still based on entering text commands. Also, the Macintosh OS managed computer memory closely for the software, something MS-DOS did not do. And, because Mac OS managed all computer memory for the application programs, you could start several programs and switch among them. Mac OS was also years ahead in I/O functions such as printer management. In MS-DOS, a program had to provide its own drivers for I/O devices; MS-DOS provided only the most rudimentary interface. On Mac OS, many I/O functions were part of the operating system.

Microsoft, however, did not stay behind for long. In 1985, Microsoft shipped an extension to its DOS operating system, called Microsoft Windows, which provided a GUI and many of the same functions as Mac OS. The first Windows was really an operating "environment" running on top of MS-DOS, made to look like a single operating system. Today's Windows is no longer based on DOS and is a full-fledged operating system.

Although Apple was six years ahead of Microsoft in offering a friendly GUI-based OS, Apple ultimately fell well behind Microsoft in sales because it chose not to license the Mac OS to outside hardware vendors.

Numerous incarnations of operating systems have come and gone since those days. Today, both Windows and Mac OS X are very similar in what they can do and how they do it; they have a wealth of features and drivers that make the original DOS look elementary. Their principal functions are unchanged, however: to provide an interface between the application programs and hardware, and to provide a user interface for basic functions, such as file and disk management.

Let's review the important pieces of operating system development history. Although pre-1980s computing history is interesting, it doesn't hold much relevance to what we do with computers today. Tables 1-1 and 1-2 show the major milestones in operating system development. The tables summarize 8-, 16-, 32-, and 64-bit operating systems. In general, a 64-bit operating system is more powerful and faster than a 32-bit system, which is more powerful and faster than a 16-bit system, and so on. You will learn more about these differences in Chapter 2, "Popular Operating Systems," and Chapter 3, "Operating System Hardware Components."

Table 1-1 Operating system releases from 1968 to 1999

Operating system	Approximate date	Bits	Comments
UNIX (Bell/AT&T)	1968	8	First widely used multiuser, multitasking operating system for minicomputers.
CP/M	1975	8	First operating system that allowed serious business work on small personal computers. VisiCalc, a spreadsheet application released in 1978, was the first business calculation program for CP/M, and to a large extent made CP/M a success.
MS-DOS	1980	16	First operating system for the very successful IBM PC family of computers. Lotus 1-2-3 was to MS-DOS in 1981 what VisiCalc was to CP/M. Also in 1981, Microsoft introduced the first version of Word for the PC.

(*continues*)