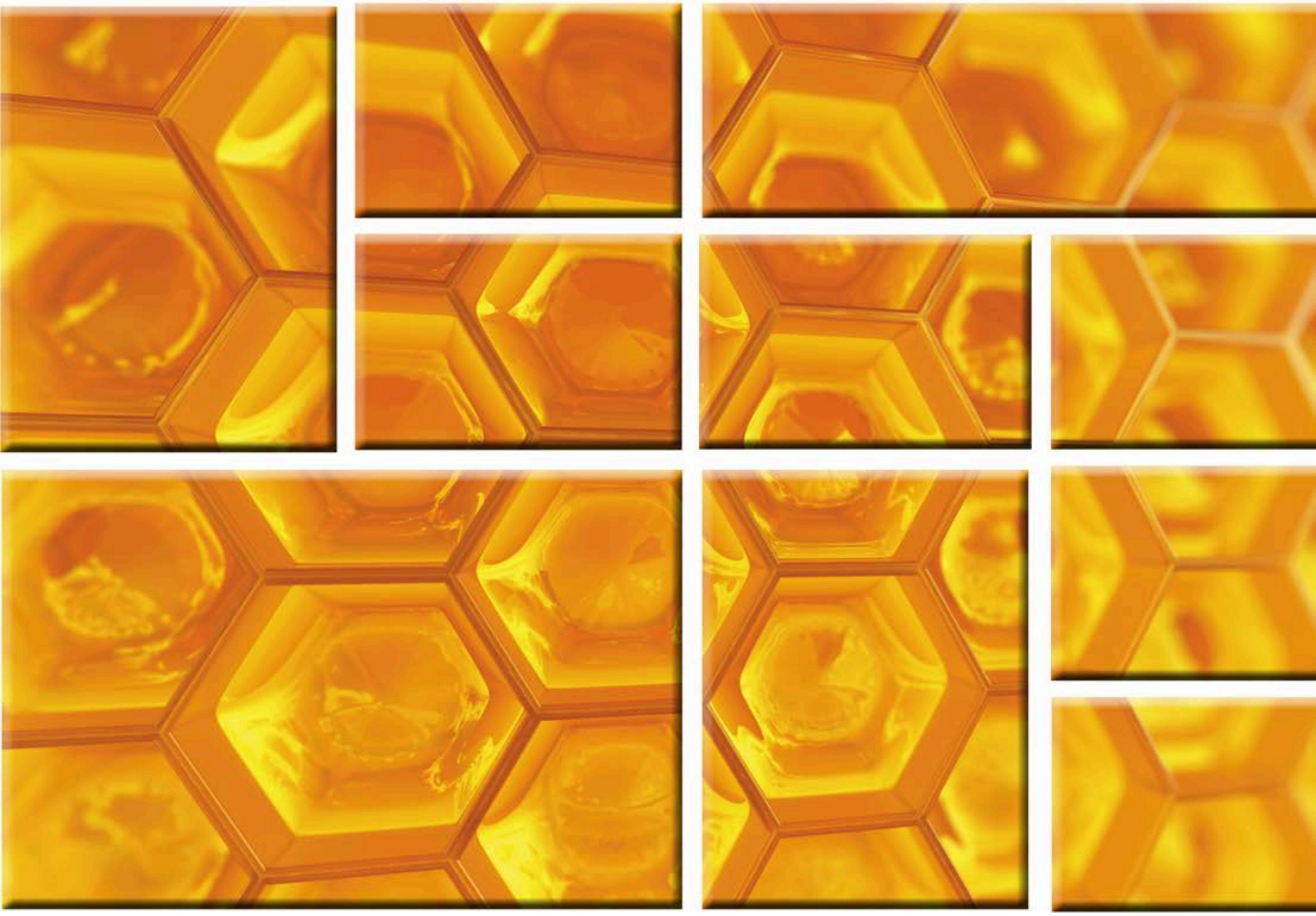


seventh edition

understanding operating systems

ann mciver mchoes and ida m. flynn

Access student data files and other study tools on **cengagebrain.com**.

For detailed instructions visit
http://s-solutions.cengage.com/ctdownloads/

Store your Data Files on a USB drive for maximum efficiency in organizing and working with the files.

Macintosh users should use a program to expand WinZip or PKZip archives.
Ask your instructor or lab coordinator for assistance.

Understanding Operating Systems

Seventh Edition

Ann McIver McHoes
Ida M. Flynn

CENGAGE
Learning®

Australia • Canada • Mexico • Singapore • Spain • United Kingdom • United States

CENGAGE Learning

Understanding Operating Systems, Seventh Edition
Ann McIver McHoes
Ida M. Flynn

Senior Product Manager: Jim Gish

Product Director: Kathleen McMahon

Senior Content Developer: Alyssa Pratt

Product Assistant: Sarah Timm

Content Project Manager: Jennifer Feltri-George

Senior Rights Acquisitions Specialist: Christine Myaskovsky

Art Director: Cheryl Pearl, GEX

Manufacturing Planner: Julio Esperas

Cover Designer: Cheryl Pearl, GEX

Cover Photos: ©Dabarti CGI/Shuttershock

Compositor: Integra

Library of Congress Control Number: 2013945460

ISBN-13: 978-1-285-09655-1

ISBN-10: 1-285-09655-X

Cengage Learning
20 Channel Center Street
Boston, MA 02210
USA

Printed in the United States of America
1 2 3 4 5 6 7 19 18 17 16 15 14 13

Dedicated to two inspiring colleagues:

Ida Moretti Flynn, award-winning teacher and a wonderful friend; her love for teaching lives on.

Bob Kleinmann, superb editor and soul mate – not in that order.

AMM

Contents

Part One **Operating Systems Concepts** **1**

Chapter 1 Introducing Operating Systems 3

Introduction 4
What Is an Operating System? 4
Operating System Software 4
Main Memory Management 6
Processor Management 7
Device Management 7
File Management 8
Network Management 8
User Interface 9
Cooperation Issues 10
Cloud Computing 11
An Evolution of Computing Hardware 11
Types of Operating Systems 13
Brief History of Operating Systems Development 16
1940s 16
1950s 16
1960s 18
1970s 18
1980s 19
1990s 20
2000s 20
2010s 22
Design Considerations 23
Conclusion 23
Key Terms 24
Interesting Searches 25
Exercises 26

Chapter 2 Memory Management: Simple Systems 29

Single-User Contiguous Scheme 30
  Fixed Partitions 31
Dynamic Partitions 34
Best-Fit and First-Fit Allocation 36
Deallocation 41
  Case 1: Joining Two Free Blocks 41
  Case 2: Joining Three Free Blocks 42
  Case 3: Deallocating an Isolated Block 43
Relocatable Dynamic Partitions 45
Conclusion 50
Key Terms 51
Interesting Searches 52
Exercises 52

Chapter 3 Memory Management: Virtual Memory Systems 59

Paged Memory Allocation 60
Demand Paging Memory Allocation 66
Page Replacement Policies and Concepts 71
  First-In First-Out 71
  Least Recently Used 73
  Clock Replacement Variation 74
  Bit Shifting Variation 75
  The Mechanics of Paging 76
  The Working Set 78
Segmented Memory Allocation 81
Segmented/Demand Paged Memory Allocation 84
Virtual Memory 87
Cache Memory 89
Conclusion 92
Key Terms 93
Interesting Searches 95
Exercises 96

Chapter 4 Processor Management 103

Overview 104
  Definitions 104
  About Multi-Core Technologies 106

	Scheduling Submanagers	107
	Process Scheduler	108
	Job and Process States	110
	Thread States	111
	Control Blocks	112
	Control Blocks and Queuing	114
	Scheduling Policies	115
	Scheduling Algorithms	116
	First-Come, First-Served	116
	Shortest Job Next	118
	Priority Scheduling	119
	Shortest Remaining Time	120
	Round Robin	122
	Multiple-Level Queues	125
	Earliest Deadline First	127
	Managing Interrupts	129
	Conclusion	130
	Key Terms	131
	Interesting Searches	134
	Exercises	134
Chapter 5	**Process Management**	**139**
	Deadlock, Livelock, and Starvation	141
	Deadlock	141
	Seven Cases of Deadlock or Livelock	142
	Necessary Conditions for Deadlock or Livelock	149
	Modeling Deadlocks	150
	Strategies for Handling Deadlocks	153
	Starvation	161
	Conclusion	163
	Key Terms	164
	Interesting Searches	165
	Exercises	165
Chapter 6	**Concurrent Processes**	**173**
	What Is Parallel Processing?	174
	Levels of Multiprocessing	176
	Introduction to Multi-Core Processors	176
	Typical Multiprocessing Configurations	177
	Master/Slave Configuration	177

	Loosely Coupled Configuration	178
	Symmetric Configuration	179
	Process Synchronization Software	180
	Test-and-Set	181
	WAIT and SIGNAL	182
	Semaphores	182
	Process Cooperation	185
	Producers and Consumers	185
	Readers and Writers	188
	Concurrent Programming	189
	Amdahl's Law	190
	Order of Operations	191
	Applications of Concurrent Programming	193
	Threads and Concurrent Programming	197
	Two Concurrent Programming Languages	198
	Ada	198
	Java	199
	Conclusion	202
	Key Terms	202
	Interesting Searches	204
	Exercises	204
Chapter 7	**Device Management**	**209**
	Types of Devices	210
	Management of I/O Requests	211
	I/O Devices in the Cloud	213
	Sequential Access Storage Media	213
	Direct Access Storage Devices	216
	Magnetic Disk Storage	216
	Access Times	218
	Optical Disc Storage	227
	CD and DVD Technology	229
	Blu-ray Disc Technology	230
	Solid State Storage	231
	Flash Memory Storage	231
	Solid State Drives	232
	Components of the I/O Subsystem	233
	Communication Among Devices	236
	RAID	239
	Level Zero	241
	Level One	242

	Level Two	243
	Level Three	243
	Level Four	244
	Level Five	244
	Level Six	245
	Nested RAID Levels	245
	Conclusion	246
	Key Terms	247
	Interesting Searches	250
	Exercises	250
Chapter 8	**File Management**	**255**
	The File Manager	256
	Interacting with the File Manager	259
	Typical Volume Configuration	260
	Introducing Subdirectories	261
	File-Naming Conventions	263
	File Organization	265
	Record Format	265
	Physical File Organization	266
	Physical Storage Allocation	269
	Contiguous Storage	270
	Noncontiguous Storage	271
	Indexed Storage	273
	Access Methods	274
	Sequential Access	274
	Direct Access	275
	Levels in a File Management System	276
	Access Control Verification Module	278
	Access Control Matrix	279
	Access Control Lists	280
	Capability Lists	281
	Data Compression	281
	Text Compression	282
	Image and Sound Compression	283
	Conclusion	283
	Key Terms	284
	Interesting Searches	286
	Exercises	286

Chapter 9 Network Organization Concepts 291

Basic Terminology 292
Network Topologies 294
Star 294
Ring 295
Bus 296
Tree 298
Hybrid 298
Network Types 299
Personal Area Network 299
Local Area Network 300
Metropolitan Area Network 300
Wide Area Network 301
Wireless Local Area Network 301
Software Design Issues 302
Addressing Conventions 302
Routing Strategies 303
Connection Models 305
Conflict Resolution 308
Transport Protocol Standards 312
OSI Reference Model 313
TCP/IP Model 316
Conclusion 319
Key Terms 320
Interesting Searches 321
Exercises 321

Chapter 10 Management of Network Functions 325

History of Networks 326
Comparison of Two Networking Systems 326
DO/S Development 330
Memory Management 330
Process Management 331
Device Management 336
File Management 339
Network Management 343
NOS Development 345
Important NOS Features 346
Major NOS Functions 346

	Conclusion	347
	Key Terms	348
	Interesting Searches	348
	Exercises	349
Chapter 11	**Security and Ethics**	**351**
	Role of the Operating System in Security	352
	System Survivability	352
	Levels of Protection	353
	Backup and Recovery	354
	Security Breaches	355
	Unintentional Modifications	355
	Intentional Attacks	355
	System Protection	362
	Antivirus Software	362
	Firewalls	364
	Authentication	365
	Encryption	367
	Password Management	368
	Password Construction	368
	Password Alternatives	370
	Social Engineering	372
	Ethics	373
	Conclusion	374
	Key Terms	375
	Interesting Searches	376
	Exercises	377
Chapter 12	**System Management**	**379**
	Evaluating an Operating System	380
	Cooperation Among Components	380
	Role of Memory Management	381
	Role of Processor Management	381
	Role of Device Management	382
	Role of File Management	384
	Role of Network Management	385
	Measuring System Performance	386
	Measurement Tools	387
	Feedback Loops	389

Patch Management 391
Patching Fundamentals 392
Software to Manage Deployment 395
Timing the Patch Cycle 395
System Monitoring 395
Conclusion 399
Key Terms 399
Interesting Searches 400
Exercises 400

Part Two Operating Systems in Practice 405

Chapter 13 UNIX Operating Systems 407

Brief History 408
The Evolution of UNIX 409
Design Goals 411
Memory Management 411
Process Management 413
Process Table Versus User Table 414
Synchronization 416
Device Management 419
Device Classifications 419
Device Drivers 421
File Management 422
File Naming Conventions 423
Directory Listings 424
Data Structures 426
User Interfaces 428
Script Files 429
Redirection 429
Pipes 431
Filters 431
Additional Commands 433
Conclusion 436
Key Terms 436
Interesting Searches 437
Exercises 437

Chapter 14	Windows Operating Systems	441
	Brief History	442
	Design Goals	444
	Extensibility	444
	Portability	444
	Reliability	445
	Compatibility	446
	Performance	446
	Memory Management	447
	User Mode Features	448
	Virtual Memory Implementation	448
	Processor Management	450
	Device Management	452
	File Management	456
	Network Management	459
	MS-NET	459
	Directory Services	460
	Security Management	461
	Security Concerns	462
	Security Terminology	463
	User Interface	464
	Conclusion	467
	Key Terms	467
	Interesting Searches	469
	Exercises	469
Chapter 15	Linux Operating Systems	473
	Brief History	474
	Design Goals	475
	Memory Management	477
	Processor Management	479
	Organization of Process Table	480
	Process Synchronization	480
	Process Management	480
	Device Management	482
	Device Classifications	482
	Device Drivers	483
	Device Classes	484

	File Management	485
	Data Structures	485
	Filename Conventions	485
	Data Structures	487
	New Versions	487
	User Interface	488
	System Monitor	489
	System Logs	489
	File Listings	490
	Setting Permissions	491
	Conclusion	492
	Key Terms	492
	Interesting Searches	493
	Exercises	493
Chapter 16	**Android Operating Systems**	**497**
	Brief History	498
	Design Goals	500
	Memory Management	501
	Processor Management	502
	Manifest, Activity, Task, and Intent	502
	Activity States	503
	Device Management	506
	Screen Requirements	506
	Battery Management	507
	File Management	508
	Security Management	509
	Permissions	509
	Device Access Security	510
	Encryption Options	512
	Bring Your Own Devices	512
	User Interface	514
	Touch Screen Controls	514
	User Interface Elements	515
	Conclusion	517
	Key Terms	517
	Interesting Searches	518
	Exercises	518

Appendix

Appendix A Algorithms 521
Appendix B ACM Code of Ethics and Professional Conduct 527

Glossary 531

Bibliography 557

Index 561

Preface

Is this book for you? In these pages, we explain a very technical subject in a not-so-technical manner, putting the concepts of operating systems into a format that many readers can quickly grasp.

For those who are new to the subject, this text demonstrates what operating systems are, what they do, how they do it, how their performance can be evaluated, and how they compare with each other. Throughout the text we describe the overall function and lead readers to additional resources where they can find more detailed information, if they so desire.

For those with more technical backgrounds, this text introduces the subject concisely, describing the complexities of operating systems without going into intricate detail. One might say this book leaves off where other operating system textbooks begin.

To do so, we've made some assumptions about our audiences. First, we assume the readers have some familiarity with computing systems. Second, we assume they have a working knowledge of an operating system and how it interacts with them. We recommend (although we don't require) that readers be familiar with at least one operating system. In the few places where, in previous editions, we used pseudocode to illustrate the inner workings of the operating systems, we have moved that code to the Appendix. In those places, we use a prose description that explains the events in familiar terms. The algorithms are still available but by moving them to the Appendix, we have simplified our explanations of some complex events.

Organization and Features

This book is structured to explain the functions of an operating system regardless of the hardware that houses it. The organization addresses a recurring problem with textbooks about technologies that continue to change—that is, the constant advances in evolving subject matter can make textbooks immediately outdated. To address this problem, we've divided the material into two parts: first, the concepts—which do not change quickly—and second, the specifics of operating systems—which change dramatically over the course of years and even months. Our goal is to give readers the ability to apply the topics intelligently, realizing that, although a command, or series of commands, used by one operating system may be different from another, their goals are the same and the functions of the operating systems are also the same.

Although it is more difficult to understand how operating systems work than to memorize the details of a single operating system, understanding general operating system concepts is a longer-lasting achievement. Such understanding also pays off in the long run because it allows one to adapt as technology changes—as, inevitably, it does. Therefore, the purpose of this book is to give computer users a solid background in the basics of operating systems, their functions and goals, and how they interact and interrelate.

Part One, the first 12 chapters, describes the theory of operating systems. It concentrates on each of the "managers" in turn and shows how they work together. Then it introduces network organization concepts, security, ethics, and management of network functions. Part Two examines actual operating systems—how they apply the theories presented in Part One and how they compare with each other.

Chapter 1 gives a brief introduction to the subject. The meat of the text begins in Chapters 2 and 3 with memory management because it is the simplest component of the operating system to explain and has historically been tied to the advances from one operating system to the next. We explain the role of the Processor Manager in Chapters 4, 5, and 6, first discussing simple systems and then expanding the discussion to include multiprocessing systems. By the time we reach device management in Chapter 7 and file management in Chapter 8, readers will have been introduced to the four main managers found in every operating system. Chapters 9 and 10 introduce basic concepts related to networking, and Chapters 11 and 12 discuss security, ethics, and some of the tradeoffs that designers consider when attempting to satisfy the needs of their user population.

Each chapter includes learning objectives, key terms, and research topics. For technically oriented readers, the exercises at the end of each chapter include problems for advanced students. Please note that some advanced exercises assume knowledge of matters not presented in the book, but they're good for those who enjoy a challenge. We expect some readers from a more general background will cheerfully pass them by.

In an attempt to bring the concepts closer to home, throughout the book we've added real-life examples to illustrate abstract concepts. However, let no one confuse our conversational style with our considerable respect for the subject matter. The subject of operating systems is a complex one and it cannot be covered completely in these few pages. Therefore, this textbook does not attempt to give an in-depth treatise of operating systems theory and applications. This is the overall view.

Part Two introduces four operating systems in the order of their first release: UNIX, Windows, Linux, and the most recent, Android. Here, each chapter discusses how one operating system applies the concepts discussed in Part One and how it compares with the others. Again, we must stress that this is a general discussion—an in-depth examination of an operating system would require details based on its current standard version, which can't be done here. We strongly suggest that readers use our discussion as a guide—a base to work from—when comparing the advantages and disadvantages of a specific operating system and supplement our work with current research that's readily available on the Internet.

The text concludes with several reference aids. Terms that are important within a chapter are listed at its conclusion as key terms. The extensive end-of-book Glossary includes brief reader-friendly definitions for hundreds of terms used in these pages. The Bibliography can guide the reader to basic research on the subject. Finally, the Appendix features pseudocode algorithms and the ACM Code of Ethics.

Not included in this text is a detailed discussion of databases and data structures, except as examples of process synchronization problems. This is because these structures only tangentially relate to operating systems and are frequently the subject of other courses. We suggest that readers begin by learning the basics as presented in the following pages before pursuing these complex subjects.

Changes to the Seventh Edition

This edition has been thoroughly updated and features many improvements over previous editions:

- A new chapter featuring the Android operating system
- New chapter spotlights on industry innovators; award-winning individuals who have propelled operating system technologies.
- Numerous new examples of operating system technology
- Updated references to the expanding influence of wireless technology
- New collection of memory and processor management pseudocode algorithms in the Appendix for those who want to understand them in greater detail.
- Enhanced discussion of patch management and system durability
- New discussions of Amdahl's law and Flynn's taxonomy
- More discussion describing the management of multiple processors
- Updated detail in the chapters that discuss UNIX, Windows, and Linux
- New homework exercises in every chapter

The MS-DOS chapter that appeared in previous editions has been retired. But, in response to faculty requests, it continues to be available in its entirety from the Cengage website so adopters can still allow students to learn the basics of this command-driven interface using a Windows emulator.

Numerous other changes throughout the text are editorial clarifications, expanded captions, and improved illustrations.

A Note for Instructors

The following supplements are available when this text is used in a classroom setting. All supplements can be downloaded from the Instructor Companion Site. Simply search for this text at sso.cengage.com. An instructor login is required.

Electronic Instructor's Manual. The Instructor's Manual that accompanies this textbook includes additional instructional material to assist in class preparation, including Sample Syllabi, Chapter Outlines, Technical Notes, Lecture Notes, Quick Quizzes, Teaching Tips, and Discussion Topics.

ExamView® Test Bank. This textbook is accompanied by ExamView, a powerful testing software package that allows instructors to create and administer printed, computer, and Internet exams. ExamView includes hundreds of questions that correspond to the topics covered in this text, enabling students to generate detailed study guides that include page references for future review.

PowerPoint Presentations. This book comes with Microsoft PowerPoint slides for each chapter. These are included as a teaching aid for classroom presentations, either to make available to students on the network for chapter review, or to be printed for classroom distribution. Instructors can add their own slides for additional topics that they introduce to the class.

Solutions. Selected solutions to Review Questions and Exercises are provided.

Order of Presentation

We have built this text with a modular construction to accommodate several presentation options, depending on the instructor's preference. For example, the syllabus can follow the chapters as listed in Chapter 1 through Chapter 12 to present the core concepts that all operating systems have in common. Using this path, students will learn about the management of memory, processors, devices, files, and networks, in that order. An alternative path might begin with Chapter 1, move next to processor management in Chapters 4 through 6, then to memory management in Chapters 2 and 3, touch on systems security and management in Chapters 11 and 12, and finally move to device and file management in Chapters 7 and 8. Because networking is often the subject of another course, instructors may choose to bypass Chapters 9 and 10, or include them for a more thorough treatment of operating systems.

We hope you find our discussion of ethics helpful in Chapter 11, which is included in response to requests by university adopters of the text who want to discuss this subject in their lectures.

In Part Two, we examine details about four specific operating systems in an attempt to show how the concepts in the first 12 chapters are applied by a specific operating system. In each case, the chapter is structured in a similar manner as the chapters in Part One. That is, they discuss the management of memory, processors, files, devices, networks, and systems. In addition, each includes an introduction to one or more user interfaces for that operating system. To illustrate the use of graphical user interfaces in UNIX systems, we include references to the Macintosh OS X operating system in the UNIX chapter.

With this edition, we have added a discussion of the Android operating system. By adding this software, specifically written for use in a mobile environment using phones and tablets, we are able to explore the challenges unique to these computing situations.

If you have suggestions for inclusion in this text, please send them along. Although we are squeezed for space, we are pleased to consider all possibilities.

Acknowledgments

Our gratitude goes to all of our friends and colleagues, who were so generous with their encouragement, advice, and support over the two decades of this publication. Special thanks go to Bob Kleinmann, Eleanor Irwin, and Roger Flynn for their assistance.

As always, thanks to those at Cengage, Brooks/Cole, and PWS Publishing who have made significant contributions to all seven editions of this text, especially Alyssa Pratt, Kallie Swanson, Mike Sugarman, and Mary Thomas Stone. In addition, the following individuals made key contributions to this edition: Jennifer Feltri-George, Content Project Manager, and Suwathiga Velayutham, Integra.

We deeply appreciate the comments of the reviewers who helped us refine this edition:

Larry Merkle, Computational Optimization Services; Michelle Parker, Indiana University—Purdue University Fort Wayne; and Kong-Cheng Wong, Governors State University.

And to the many students and instructors who have sent helpful comments and suggestions since publication of the first edition in 1991, we thank you. Please keep them coming.

Ann McIver McHoes, mchoesa@duq.edu

Ida M. Flynn

Part One

Operating Systems Concepts

"Dost thou not see ... the bees working together to put in order theirseveral parts of the universe?"

—Marcus Aurelius Antoninus (121–180)

Like honey bees, the core mechanisms of operating systems must work together to manage the operating system's memory, processing capability, devices and peripherals, files, and networks—and do so in an appropriate and secure fashion. Here in Part One, we present an overview of these operating systems essentials.

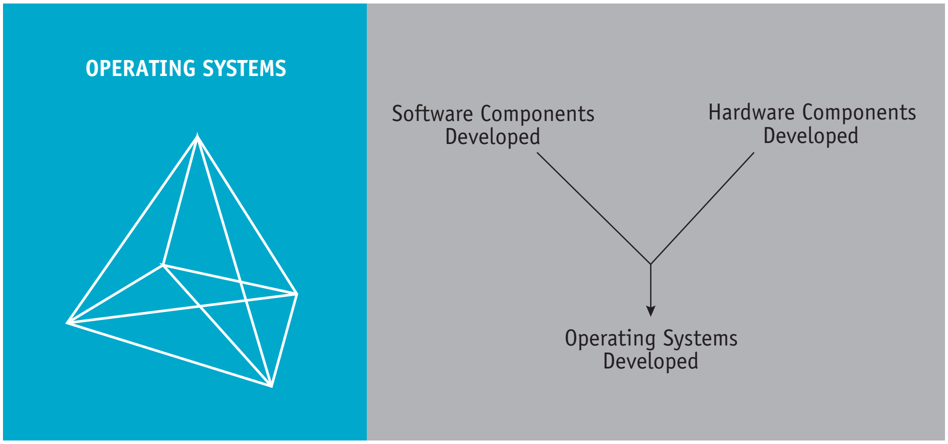
- Chapter 1 introduces the subject.
- Chapters 2 and 3 discuss main memory management.
- Chapters 4 through 6 cover processor management.
- Chapter 7 concentrates on device management.
- Chapter 8 is devoted to file management.
- Chapters 9 and 10 briefly review networks.
- Chapter 11 discusses system security.
- Chapter 12 explores system management.

In Part Two (Chapters 13 through 16), we look at four specific operating systems and how they apply the concepts presented here in Part One.

Throughout our discussion of this very technical subject, we try to include definitions of terms that might be unfamiliar to you, but it isn't always possible to describe a function and define the technical terms while keeping the explanation clear. Therefore, we've put the key terms with definitions at the end of each chapter and again in the glossary at the end of the text. Items listed in the Key Terms are shown in **boldface** the first time they are mentioned significantly.

Throughout the book we keep our descriptions and examples as simple as possible to introduce you to the system's complexities without getting bogged down in technical detail. Therefore, remember that for almost every topic explained in the following pages, there's much more information available for you to study. Our goal is to introduce you to the subject and to encourage you to pursue your personal interests using other sources. Enjoy.

Chapter 1 Introducing Operating Systems



> *"I think there is a world market for maybe five computers."*
>
> —attributed to Thomas J. Watson (1874–1956; chairman of IBM 1949–1956)

Learning Objectives

After completing this chapter, you should be able to describe:

- Innovations in operating systems development
- The basic role of an operating system
- The major operating system software subsystem managers and their functions
- The types of machine hardware on which operating systems run
- The differences among batch, interactive, real-time, hybrid, and embedded operating systems
- Design considerations of operating systems designers

Introduction

To understand an operating system is to begin to understand the workings of an entire computer system, because the operating system software manages each and every piece of hardware and software. In the pages that follow, we explore what operating systems are, how they work, what they do, and why.

This chapter briefly describes the workings of operating systems on the simplest scale. The following chapters explore each component in more depth and show how its function relates to the other parts of the operating system. In other words, you see how the pieces work together harmoniously to keep the computer system working smoothly.

What Is an Operating System?

A computer system typically consists of software (programs) and hardware (the tangible machine and its electronic components). The operating system software is the chief piece of software, the portion of the computing system that manages all of the hardware and all of the other software. To be specific, it controls every file, every device, every section of main memory, and every moment of processing time. It controls who can use the system and how. In short, the operating system is the boss.

Therefore, each time the user sends a command, the operating system must make sure that the command is executed, or if it's not executed, it must arrange for the user to get a message explaining the error. Remember: this doesn't necessarily mean that the operating system executes the command or sends the error message—but it does control the parts of the system that do.

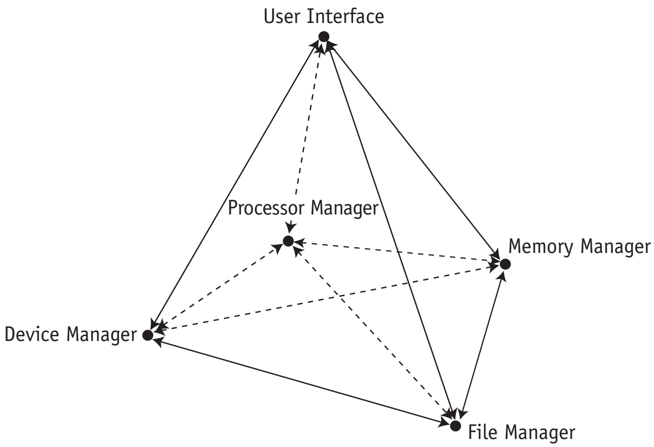
Operating System Software

The pyramid shown in Figure 1.1 is an abstract representation of the operating system in its simplest form and demonstrates how its major components typically work together.

At the base of the pyramid are the four essential managers of every major operating system: the **Memory Manager, Processor Manager, Device Manager,** and **File Manager.** These managers and their interactions are discussed in detail in Chapters 1 through 8 of this book. Each manager works closely with the other managers as each one performs its unique role. At the top of the pyramid is the User Interface, which allows the user to issue commands to the operating system. Because this component has specific elements, in both form and function, it is often very different from one operating system to the next—sometimes even between different versions of the same operating system.

(figure 1.1)

This pyramid represents an operating system on a stand-alone computer unconnected to a network. It shows the four subsystem managers and the user interface.

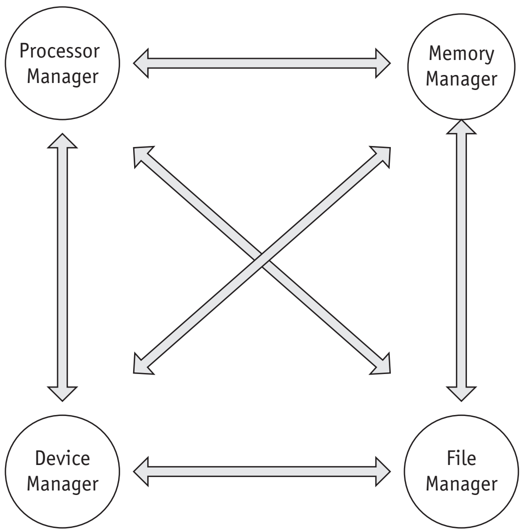


Regardless of the size or configuration of the system, the four managers illustrated in Figure 1.2 must, at a minimum, perform the following tasks while collectively keeping the system working smoothly:

- Monitor the system's resources
- Enforce the policies that determine what component gets what resources, when, and how much
- Allocate the resources when appropriate
- Deallocate the resources when appropriate

(figure 1.2)

Each manager at the base of the pyramid takes responsibility for its own tasks while working harmoniously with every other manager.



For example, the Memory Manager must keep track of the status of the computer system's main memory space, allocate the correct amount of it to incoming processes, and deallocate that space when appropriate—all while enforcing the policies that were established by the designers of the operating system.

An additional management task, networking, was not always an integral part of operating systems. Today the vast majority of major operating systems incorporate a **Network Manager** to coordinate the services required for multiple systems to work cohesively together. For example, the Network Manager must coordinate the workings of the networked resources, which might include shared access to memory space, processors, printers, databases, monitors, applications, and more. This can be a complex balancing act as the number of resources increases, as it often does.

Main Memory Management

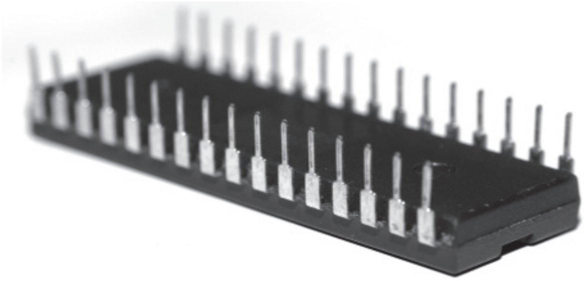
The Memory Manager (the subject of Chapters 2 and 3) is in charge of main memory, widely known as **RAM** (short for random access memory). The Memory Manager checks the validity of each request for memory space, and if it is a legal request, allocates a portion of memory that isn't already in use. If the memory space becomes fragmented, this manager might use policies established by the operating systems designers to reallocate memory to make more useable space available for other jobs that are waiting. Finally, when the job or process is finished, the Memory Manager deallocates its allotted memory space.

A key feature of RAM chips—the hardware that comprises computer memory—is that they depend on the constant flow of electricity to hold data. When the power fails or is turned off, the contents of RAM is wiped clean. This is one reason why computer system designers attempt to build elegant shutdown procedures, so the contents of RAM can be stored on a nonvolatile device, such as a hard drive, before the main memory chips lose power during computer shutdown.

RAM is the computer's main memory and is sometimes called "primary storage" to distinguish it from "secondary storage," where data is stored on hard drives or other devices.

A critical responsibility of the Memory Manager is to protect all of the space in main memory, particularly that occupied by the operating system itself—it can't allow any part of the operating system to be accidentally or intentionally altered because that would lead to instability or a system crash.

Another kind of memory that's critical when the computer is powered on is **Read-Only Memory** (often shortened to **ROM**), shown in Figure 1.3. This ROM chip holds software called **firmware**, the programming code that is used to start the computer and perform other necessary tasks. To put it in simplest form, it describes in prescribed steps when and how to load each piece of the operating system after the power is turned on and until the computer is ready for use. The contents of the ROM chip are nonvolatile, meaning that they are not erased when the power is turned off, unlike the contents of RAM.



(figure 1.3)

A computer's relatively small ROM chip contains the firmware (unchanging software) that prescribes system initialization when the system powers on.

Processor Management

The Processor Manager (discussed in Chapters 4 through 6) decides how to allocate the central processing unit (CPU); an important function of the Processor Manager is to keep track of the status of each job, process, thread, and so on. We will discuss all of these in the chapters that follow, but for this overview, let's limit our discussion to **processes** and define them as a program's "instance of execution." A simple example could be a request to solve a mathematical equation: this would be a single job consisting of several processes, with each process performing a part of the overall equation.

The Processor Manager is required to monitor the computer's CPU to see if it's busy executing a process or sitting idle as it waits for some other command to finish execution. Generally, systems are more efficient when their CPUs are kept busy. The Processor Manager handles each process's transition, from one state of execution to another, as it moves from the starting queue, through the running state, and finally to the finish line (where it then tends to the next process). Therefore, this manager can be compared to a traffic controller. When the process is finished, or when the maximum amount of computation time has expired, the Processor Manager reclaims the CPU so it can allocate it to the next waiting process. If the computer has multiple CPUs, as in a multicore system, the Process Manager's responsibilities are greatly complicated.

Device Management

A flash memory device is an example of secondary storage because it doesn't lose data when its power is turned off.

The Device Manager (the subject of Chapter 7) is responsible for connecting with every device that's available on the system and for choosing the most efficient way to allocate each of these printers, ports, disk drives, and more, based on the device scheduling policies selected by the designers of the operating system.

Good device management requires that this part of the operating system uniquely identify each device, start its operation when appropriate, monitor its progress, and finally deallocate the device to make the operating system available to the next waiting process. This isn't as easy as it sounds because of the exceptionally wide range of devices that can be attached to any system. For example, let's say you're adding a printer to your system. There are several kinds of printers commonly available (laser, inkjet, inkless thermal, etc.) and they're made by manufacturers that number in the hundreds or thousands. To complicate things, some devices can be shared, while some can be used by only one user or one job at a time. Designing an operating system to manage such a wide range of printers (as well as monitors, keyboards, pointing devices, disk drives, cameras, scanners, and so on) is a daunting task. To do so, each device has its own software, called a **device driver**, which contains the detailed instructions required by the operating system to start that device, allocate it to a job, use the device correctly, and deallocate it when it's appropriate.

File Management

The File Manager (described in Chapter 8), keeps track of every file in the system, including data files, program files, utilities, compilers, applications, and so on. By following the access policies determined by the system designers, the File Manager enforces restrictions on who has access to which files. Many operating systems allow authorized individuals to change those permissions and restrictions. The File Manager also controls the range of actions that each user is allowed to perform with files after they access them. For example, one user might have read-only access to a critical database, while the systems administrator might hold read-and-write access and the authority to create and delete files in the same database. Access control is a key part of good file management and is tightly coupled with system security software.

When the File Manager allocates space on a secondary storage device (such as a hard drive, flash drive, archival device, and so on), it must do so knowing the technical requirements of that device. For example, if it needs to store an archival copy of a large file, it needs to know if the device stores it more efficiently as one large block or in several smaller pieces that are linked through an index. This information is also necessary for the file to be retrieved correctly later. Later, if this large file must be modified after it has been stored, the File Manager must be capable of making those modifications accurately and as efficiently as possible.

Network Management

Operating systems with networking capability have a fifth essential manager called the Network Manager (the subject of Chapters 9 and 10) that provides a convenient way for authorized users to share resources. To do so, this manager must take overall responsibility for every aspect of network connectivity, including the requirements of the available devices as well as files, memory space, CPU capacity, transmission connections, and types of encryption (if necessary). Networks with many available resources require management of a vast range of alternative elements, which enormously complicates the tasks required to add network management capabilities.

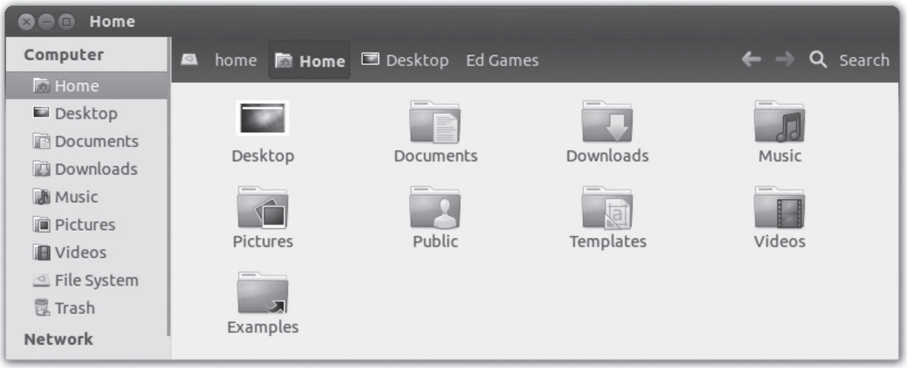
Networks can range from a small wireless system that connects a game system to the Internet, to a private network for a small business, to one that connects multiple computer systems, devices, and mobile phones to the Internet. Regardless of the size and complexity of the network, these operating systems must be prepared to properly manage the available memory, CPUs, devices, and files.

User Interface

The user interface—the portion of the operating system that users interact with directly—is one of the most unique components of an operating system. Two primary types are the **graphical user interface** (GUI) shown in Figure 1.4 and the **command line interface.** The GUI relies on input from a pointing device such as a mouse or your finger. Specific menu options, desktops, and formats often vary widely from one operating system to another (and sometimes from one version to another).

The alternative to a GUI is a command line interface, which responds to specific commands typed on a keyboard and displayed on the monitor, as shown in Figure 1.5. These interfaces accept typed commands and offer skilled users powerful additional control because typically the commands can be linked together (concatenated) to perform complex tasks with a single multifunctional command that would require many mouse clicks to duplicate using a graphical interface.

While a command structure offers powerful functionality, it has strict requirements for every command: each must be typed accurately, each must be formed in correct syntax, and combinations of commands must be assembled correctly. In addition, users need to know how to recover gracefully from any errors they encounter. These command line interfaces were once standard for operating systems and are still favored by power users but have largely been supplemented with simple, forgiving graphical user interfaces.



(figure 1.4)

An example of the graphical user interface (GUI) for Ubuntu Linux.