The background of the book cover is a colorful illustration of a prehistoric scene. In the foreground, three large, long-necked sauropod dinosaurs are depicted. The one on the right is the largest and most prominent, facing left. Behind it and to the left are two smaller ones. In the lower-left corner, there are two crocodiles near a body of water. The background features a range of mountains, including a prominent volcano with a plume of white smoke rising from its peak. The sky is a pale blue-green. The overall style is that of a classic children's book illustration.

**SILBERSCHATZ**

**GALVIN**

**GAGNE**

**OPERATING  
SYSTEM  
CONCEPTS**

**SIXTH EDITION**



# CONTENTS

---

## PART ONE ■ OVERVIEW

### Chapter 1 Introduction

- ✓ 1.1 What Is an Operating System? 3
- ✓ 1.2 Mainframe Systems 7
- 1.3 Desktop Systems 11
- 1.4 Multiprocessor Systems 12
- 1.5 Distributed Systems 14
- 1.6 Clustered Systems 16
- 1.7 Real-Time Systems 17

- 1.8 Handheld Systems 19
- 1.9 Feature Migration 20
- 1.10 Computing Environments 21
- 1.11 Summary 23
- Exercises 24
- Bibliographical Notes 25

### Chapter 2 Computer-System Structures

- 2.1 Computer-System Operation 27
- 2.2 I/O Structure 30
- 2.3 Storage Structure 34
- 2.4 Storage Hierarchy 38
- 2.5 Hardware Protection 42

- 2.6 Network Structure 48
- 2.7 Summary 51
- Exercises 52
- Bibliographical Notes 54

## Chapter 3 Operating-System Structures

3.1 System Components	55
3.2 Operating-System Services	61
3.3 System Calls	63
3.4 System Programs	72
3.5 System Structure	74
3.6 Virtual Machines	80

3.7 System Design and Implementation	85
3.8 System Generation	88
3.9 Summary	89
Exercises	90
Bibliographical Notes	92

## PART TWO ■ PROCESS MANAGEMENT

### Chapter 4 Processes

4.1 Process Concept	95
4.2 Process Scheduling	99
4.3 Operations on Processes	103
4.4 Cooperating Processes	107
4.5 Interprocess Communication	109

4.6 Communication in Client-Server Systems	117
4.7 Summary	126
Exercises	127
Bibliographical Notes	128

### Chapter 5 Threads

5.1 Overview	129
5.2 Multithreading Models	132
5.3 Threading Issues	135
5.4 Pthreads	139
5.5 Solaris 2 Threads	141
5.6 Window 2000 Threads	143

5.7 Linux Threads	144
5.8 Java Threads	145
5.9 Summary	147
Exercises	147
Bibliographical Notes	148

### Chapter 6 CPU Scheduling

6.1 Basic Concepts	151
6.2 Scheduling Criteria	155
6.3 Scheduling Algorithms	157
6.4 Multiple-Processor Scheduling	169
6.5 Real-Time Scheduling	170

6.6 Algorithm Evaluation	172
6.7 Process Scheduling Models	177
6.8 Summary	184
Exercises	185
Bibliographical Notes	187



## Chapter 7 Process Synchronization

- 7.1 Background 189
- 7.2 The Critical-Section Problem 191
- 7.3 Synchronization Hardware 197
- 7.4 Semaphores 201
- 7.5 Classic Problems of Synchronization 206
- 7.6 Critical Regions 211
- 7.7 Monitors 216
- 7.8 OS Synchronization 223
- 7.9 Atomic Transactions 225
- 7.10 Summary 235
- Exercises 236
- Bibliographical Notes 240

## Chapter 8 Deadlocks

- 8.1 System Model 243
- 8.2 Deadlock Characterization 245
- 8.3 Methods for Handling Deadlocks 248
- 8.4 Deadlock Prevention 250
- 8.5 Deadlock Avoidance 253
- 8.6 Deadlock Detection 260
- 8.7 Recovery from Deadlock 264
- 8.8 Summary 266
- Exercises 266
- Bibliographical Notes 270

## PART THREE ■ STORAGE MANAGEMENT

### Chapter 9 Memory Management

- 1 Background 273
- 2 Swapping 280
- 3 Contiguous Memory Allocation 283
- 4 Paging 287
- 5 Segmentation 303
- 9.6 Segmentation with Paging 309
- 9.7 Summary 312
- Exercises 313
- Bibliographical Notes 316

### Chapter 10 Virtual Memory

- 0.1 Background 317
- 0.2 Demand Paging 320
- 0.3 Process Creation 328
- 0.4 Page Replacement 330
- 0.5 Allocation of Frames 344
- 0.6 Thrashing 348
- 10.7 Operating-System Examples 353
- 10.8 Other Considerations 356
- 10.9 Summary 363
- Exercises 364
- Bibliographical Notes 369

## Chapter 11 File-System Interface

- 11.1 File Concept 371
- 11.2 Access Methods 379
- 11.3 Directory Structure 383
- 11.4 File-System Mounting 393
- 11.5 File Sharing 395

- 11.6 Protection 402
- 11.7 Summary 406
- Exercises 407
- Bibliographical Notes 409

## Chapter 12 File-System Implementation

- 12.1 File-System Structure 411
- 12.2 File-System Implementation 413
- 12.3 Directory Implementation 420
- 12.4 Allocation Methods 421
- 12.5 Free-Space Management 430
- 12.6 Efficiency and Performance 433

- 12.7 Recovery 437
- 12.8 Log-Structured File System 439
- 12.9 NFS 441
- 12.10 Summary 448
- Exercises 449
- Bibliographical Notes 451

# PART FOUR ■ I/O SYSTEMS

## Chapter 13 I/O Systems

- 13.1 Overview 455
- 13.2 I/O Hardware 456
- 13.3 Application I/O Interface 466
- 13.4 Kernel I/O Subsystem 472
- 13.5 Transforming I/O to Hardware Operations 478

- 13.6 STREAMS 481
- 13.7 Performance 483
- 13.8 Summary 487
- Exercises 487
- Bibliographical Notes 488

## Chapter 14 Mass-Storage Structure

- 14.1 Disk Structure 491
- 14.2 Disk Scheduling 492
- 14.3 Disk Management 498
- 14.4 Swap-Space Management 502
- 14.5 RAID Structure 505

- 14.7 Stable-Storage Implementation 514
- 14.8 Tertiary-Storage Structure 516
- 14.9 Summary 526
- Exercises 528
- Bibliographical Notes 529



## PART FIVE ■ DISTRIBUTED SYSTEMS

### Chapter 15 Distributed System Structures

- 15.1 Background 539
- 15.2 Topology 546
- 15.3 Network Types 548
- 15.4 Communication 551
- 15.5 Communication Protocols 558
- 15.6 Robustness 562
- 15.7 Design Issues 564
- 15.8 An Example: Networking 566
- 15.9 Summary 568
- Exercises 569
- Bibliographical Notes 571

### Chapter 16 Distributed File Systems

- 16.1 Background 573
- 16.2 Naming and Transparency 575
- 16.3 Remote File Access 579
- 16.4 Stateful Versus Stateless Service 583
- 16.5 File Replication 585
- 16.6 An Example: AFS 586
- 16.7 Summary 591
- Exercises 592
- Bibliographical Notes 593

### Chapter 17 Distributed Coordination

- 17.1 Event Ordering 595
- 17.2 Mutual Exclusion 598
- 17.3 Atomicity 601
- 17.4 Concurrency Control 605
- 17.5 Deadlock Handling 610
- 17.6 Election Algorithms 618
- 17.7 Reaching Agreement 620
- 17.8 Summary 623
- Exercises 624
- Bibliographical Notes 625

## PART SIX ■ PROTECTION AND SECURITY

### Chapter 18 Protection

- 18.1 Goals of Protection 629
- 18.2 Domain of Protection 630
- 18.3 Access Matrix 636
- 18.4 Implementation of Access Matrix 640
- 18.6 Capability-Based Systems 645
- 18.7 Language-Based Protection 648
- 18.8 Summary 654
- Exercises 655
- Bibliographical Notes 656

## Chapter 19 Security

- 19.1 The Security Problem 657
- 19.2 User Authentication 659
- 19.3 Program Threats 663
- 19.4 System Threats 666
- 19.5 Securing Systems and Facilities 671
- 19.6 Intrusion Detection 674
- 19.7 Cryptography 680

- 19.8 Computer-Security Classifications 686
- 19.9 An Example: Windows NT 687
- 19.10 Summary 689
- Exercises 690
- Bibliographical Notes 691

## PART SEVEN ■ CASE STUDIES

### Chapter 20 The Linux System

- 20.1 History 695
- 20.2 Design Principles 700
- 20.3 Kernel Modules 703
- 20.4 Process Management 707
- 20.5 Scheduling 711
- 20.6 Memory Management 716
- 20.7. File Systems 724

- 20.8 Input and Output 729
- 20.9 Interprocess Communication 732
- 20.10 Network Structure 734
- 20.11 Security 737
- 20.12 Summary 739
- Exercises 740
- Bibliographical Notes 741

### Chapter 21 Windows 2000

- 21.1 History 743
- 21.2 Design Principles 744
- 21.3 System Components 746
- 21.4 Environmental Subsystems 763
- 21.5 File System 766

- 21.6 Networking 774
- 21.7 Programmer Interface 780
- 21.8 Summary 787
- Exercises 787
- Bibliographical Notes 788

### Chapter 22 Historical Perspective

- 22.1 Early Systems 789
- 22.2 Atlas 796
- 22.3 XDS-940 797

- 22.6 CTSS 800
- 22.7 MULTICS 800
- 22.8 OS/360 801



**Appendix A The FreeBSD System (contents online)**

- |                          |      |                                |      |
|--------------------------|------|--------------------------------|------|
| A.1 History              | A807 | A.7 File System                | A834 |
| A.2 Design Principles    | A813 | A.8 I/O System                 | A842 |
| A.3 Programmer Interface | A815 | A.9 Interprocess Communication | A846 |
| A.4 User Interface       | A823 | A.10 Summary                   | A852 |
| A.5 Process Management   | A827 | Exercises                      | A852 |
| A.6 Memory Management    | A831 | Bibliographical Notes          | A853 |

**Appendix B The Mach System (contents online)**

- |                                |      |                          |      |
|--------------------------------|------|--------------------------|------|
| B.1 History                    | A855 | B.7 Programmer Interface | A880 |
| B.2 Design Principles          | A857 | B.8 Summary              | A881 |
| B.3 System Components          | A858 | Exercises                | A882 |
| B.4 Process Management         | A862 | Bibliographical Notes    | A883 |
| B.5 Interprocess Communication | A868 | Credits                  | A885 |
| B.6 Memory Management          | A874 |                          |      |

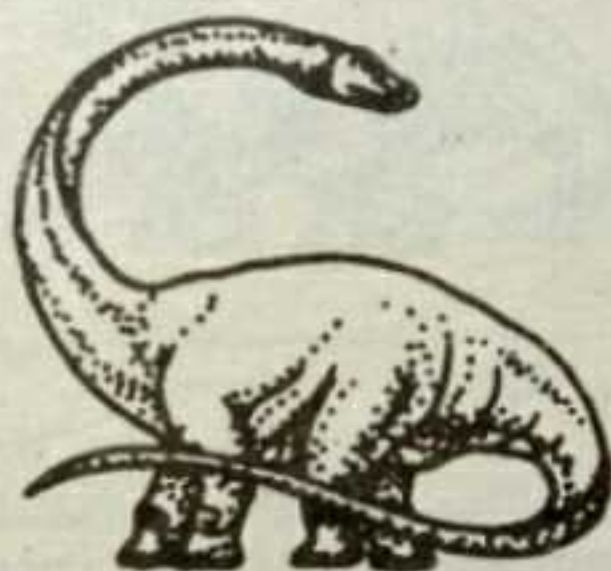
**Appendix C The Nachos System (contents online)**

- |                                |      |                       |      |
|--------------------------------|------|-----------------------|------|
| C.1 Overview                   | A888 | C.5 Conclusions       | A900 |
| C.2 Nachos Software Structure  | A890 | Bibliographical Notes | A901 |
| C.3 Sample Assignments         | A893 | Credits               | A902 |
| C.4 Obtaining a Copy of Nachos | A898 |                       |      |

**Bibliography 807****Credits 837****Index 839**



# Part One



## OVERVIEW

---

An *operating system* is a program that acts as an intermediary between the user of a computer and the computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner.

We trace the development of operating systems from the first hands-on systems, through multiprogrammed and time-shared systems, to current handheld and real-time systems. Understanding the evolution of operating systems gives us an appreciation for what an operating system does and how it does it.

The operating system must ensure the correct operation of the computer system. To prevent user programs from interfering with the proper operation of the system, the hardware must provide appropriate mechanisms. We describe the basic computer architecture that makes it possible to write a correct operating system.

The operating system provides certain services to programs and to the users of those programs in order to make their tasks easier. The services differ from one operating system to another, but we identify and explore some common classes of these services.