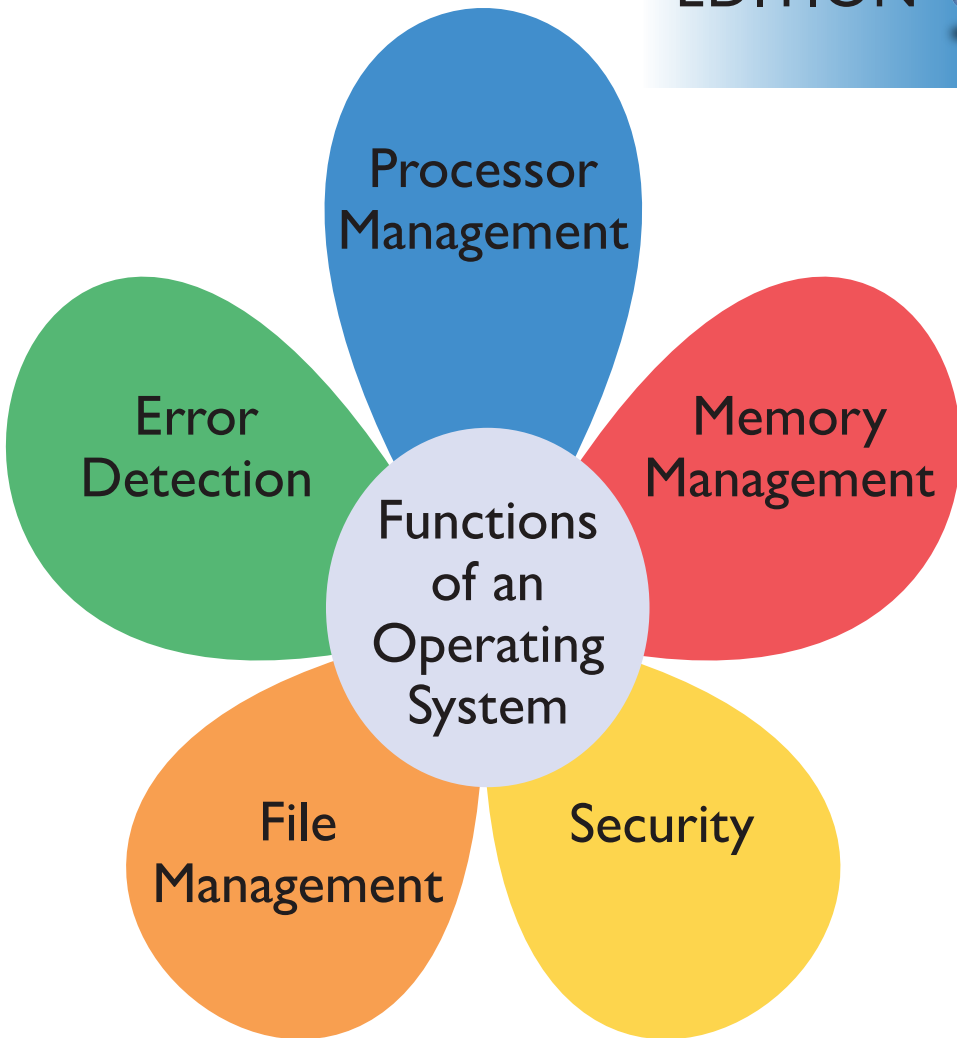


GLOBAL
EDITION



Modern Operating Systems

FIFTH EDITION

Tanenbaum • Bos



**MODERN
OPERATING SYSTEMS**

**FIFTH EDITION
GLOBAL EDITION**

Modern Operating Systems, Global Edition

Table of Contents

Cover

Half Title

Title Page

Copyright

Dedication

Contents

Preface

Chapter 1. Introduction

1.1 What Is An Operating System?

1.1.1 The Operating System as an Extended Machine

1.1.2 The Operating System as a Resource Manager

1.2 History Of Operating Systems

1.2.1 The First Generation (1945-1955): Vacuum Tubes

1.2.2 The Second Generation (1955-1965): Transistors and Batch Systems

1.2.3 The Third Generation (1965-1980): ICs and Multiprogramming

1.2.4 The Fourth Generation (1980-Present): Personal Computers

1.2.5 The Fifth Generation (1990-Present): Mobile Computers

1.3 Computer Hardware Review

1.3.1 Processors

1.3.2 Memory

1.3.3 Nonvolatile Storage

1.3.4 I/O Devices

Table of Contents

1.3.5 Buses

1.3.6 Booting the Computer

1.4 The Operating System Zoo

1.4.1 Mainframe Operating Systems

1.4.2 Server Operating Systems

1.4.3 Personal Computer Operating Systems

1.4.4 Smartphone and Handheld Computer Operating Systems

1.4.5 The Internet of Things and Embedded Operating Systems

1.4.6 Real-Time Operating Systems

1.4.7 Smart Card Operating Systems

1.5 Operating System Concepts

1.5.1 Processes

1.5.2 Address Spaces

1.5.3 Files

1.5.4 Input/Output

1.5.5 Protection

1.5.6 The Shell

1.5.7 Ontogeny Recapitulates Phylogeny

1.6 System Calls

1.6.1 System Calls for Process Management

1.6.2 System Calls for File Management

1.6.3 System Calls for Directory Management

1.6.4 Miscellaneous System Calls

1.6.5 The Windows API

1.7 Operating System Structure

1.7.1 Monolithic Systems

1.7.2 Layered Systems

1.7.3 Microkernels

1.7.4 Client-Server Model

Table of Contents

1.7.5 Virtual Machines

1.7.6 Exokernels and Unikernels

1.8 The World According To C

1.8.1 The C Language

1.8.2 Header Files

1.8.3 Large Programming Projects

1.8.4 The Model of Run Time

1.9 Research On Operating Systems

1.10 Outline Of The Rest Of This Book

1.11 Metric Units

1.12 Summary

Chapter 2. Processes And Threads

2.1 Processes

2.1.1 The Process Model

2.1.2 Process Creation

2.1.3 Process Termination

2.1.4 Process Hierarchies

2.1.5 Process States

2.1.6 Implementation of Processes

2.1.7 Modeling Multiprogramming

2.2 Threads

2.2.1 Thread Usage

2.2.2 The Classical Thread Model

2.2.3 POSIX Threads

2.2.4 Implementing Threads in User Space

2.2.5 Implementing Threads in the Kernel

2.2.6 Hybrid Implementations

2.2.7 Making Single-Threaded Code Multithreaded

2.3 Event-Driven Servers

Table of Contents

2.4 Synchronization And Interprocess Communication

- 2.4.1 Race Conditions
- 2.4.2 Critical Regions
- 2.4.3 Mutual Exclusion with Busy Waiting
- 2.4.4 Sleep and Wakeup
- 2.4.5 Semaphores
- 2.4.6 Mutexes
- 2.4.7 Monitors
- 2.4.8 Message Passing
- 2.4.9 Barriers
- 2.4.10 Priority Inversion
- 2.4.11 Avoiding Locks: Read-Copy-Update

2.5 Scheduling

- 2.5.1 Introduction to Scheduling
- 2.5.2 Scheduling in Batch Systems
- 2.5.3 Scheduling in Interactive Systems
- 2.5.4 Scheduling in Real-Time Systems
- 2.5.5 Policy Versus Mechanism
- 2.5.6 Thread Scheduling

2.6 Research On Processes And Threads

2.7 Summary

Chapter 3. Memory Management

3.1 No Memory Abstraction

- 3.1.1 Running Multiple Programs Without a Memory Abstraction

3.2 A Memory Abstraction: Address Spaces

- 3.2.1 The Notion of an Address Space
- 3.2.2 Swapping
- 3.2.3 Managing Free Memory

3.3 Virtual Memory

Table of Contents

3.3.1 Paging

3.3.2 Page Tables

3.3.3 Speeding Up Paging

3.3.4 Page Tables for Large Memories

3.4 Page Replacement Algorithms

3.4.1 The Optimal Page Replacement Algorithm

3.4.2 The Not Recently Used Page Replacement Algorithm

3.4.3 The First-In, First-Out (FIFO) Page Replacement Algorithm

3.4.4 The Second-Chance Page Replacement Algorithm

3.4.5 The Clock Page Replacement Algorithm

3.4.6 The Least Recently Used (LRU) Page Replacement Algorithm

3.4.7 Simulating LRU in Software

3.4.8 The Working Set Page Replacement Algorithm

3.4.9 The WSClock Page Replacement Algorithm

3.4.10 Summary of Page Replacement Algorithms

3.5 Design Issues For Paging Systems

3.5.1 Local versus Global Allocation Policies

3.5.2 Load Control

3.5.3 Cleaning Policy

3.5.4 Page Size

3.5.5 Separate Instruction and Data Spaces

3.5.6 Shared Pages

3.5.7 Shared Libraries

3.5.8 Mapped Files

3.6 Implementation Issues

3.6.1 Operating System Involvement with Paging

3.6.2 Page Fault Handling

3.6.3 Instruction Backup

3.6.4 Locking Pages in Memory

Table of Contents

3.6.5 Backing Store

3.6.6 Separation of Policy and Mechanism

3.7 Segmentation

3.7.1 Implementation of Pure Segmentation

3.7.2 Segmentation with Paging: MULTICS

3.7.3 Segmentation with Paging: The Intel x86

3.8 Research On Memory Management

3.9 Summary

Chapter 4. File Systems

4.1 Files

4.1.1 File Naming

4.1.2 File Structure

4.1.3 File Types

4.1.4 File Access

4.1.5 File Attributes

4.1.6 File Operations

4.1.7 An Example Program Using File-System Calls

4.2 Directories

4.2.1 Single-Level Directory Systems

4.2.2 Hierarchical Directory Systems

4.2.3 Path Names

4.2.4 Directory Operations

4.3 File-System Implementation

4.3.1 File-System Layout

4.3.2 Implementing Files

4.3.3 Implementing Directories

4.3.4 Shared Files

4.3.5 Log-Structured File Systems

4.3.6 Journaling File Systems

Table of Contents

4.3.7 Flash-based File Systems

4.3.8 Virtual File Systems

4.4 File-System Management And Optimization

4.4.1 Disk-Space Management

4.4.2 File-System Backups

4.4.3 File-System Consistency

4.4.4 File-System Performance

4.4.5 Defragmenting Disks

4.4.6 Compression and Deduplication

4.4.7 Secure File Deletion and Disk Encryption

4.5 Example File Systems

4.5.1 The MS-DOS File System

4.5.2 The UNIX V7 File System

4.6 Research On File Systems

4.7 Summary

Chapter 5. Input/Output

5.1 Principles Of I/O Hardware

5.1.1 I/O Devices

5.1.2 Device Controllers

5.1.3 Memory-Mapped I/O

5.1.4 Direct Memory Access

5.1.5 Interrupts Revisited

5.2 Principles Of I/O Software

5.2.1 Goals of the I/O Software

5.2.2 Programmed I/O

5.2.3 Interrupt-Driven I/O

5.2.4 I/O Using DMA

5.3 I/O Software Layers

Table of Contents

5.3.1 Interrupt Handlers

5.3.2 Device Drivers

5.3.3 Device-Independent I/O Software

5.3.4 User-Space I/O Software

5.4 Mass Storage: Disk And SSD

5.4.1 Magnetic Disks

5.4.2 Solid State Drives (SSDs)

5.4.3 RAID

5.5 Clocks

5.5.1 Clock Hardware

5.5.2 Clock Software

5.5.3 Soft Timers

5.6 User Interfaces: Keyboard, Mouse, & Monitor

5.6.1 Input Software

5.6.2 Output Software

5.7 Thin Clients

5.8 Power Management

5.8.1 Hardware Issues

5.8.2 Operating System Issues

5.8.3 Application Program Issues

5.9 Research On Input/Output

5.10 Summary

Chapter 6. Deadlocks

6.1 Resources

6.1.1 Preemptable and Nonpreemptable Resources

6.1.2 Resource Acquisition

6.1.3 The Dining Philosophers Problem

6.2 Introduction To Deadlocks

Table of Contents

6.2.1 Conditions for Resource Deadlocks

6.2.2 Deadlock Modeling

6.3 The Ostrich Algorithm

6.4 Deadlock Detection And Recovery

6.4.1 Deadlock Detection with One Resource of Each Type

6.4.2 Deadlock Detection with Multiple Resources of Each Type

6.4.3 Recovery from Deadlock

6.5 Deadlock Avoidance

6.5.1 Resource Trajectories

6.5.2 Safe and Unsafe States

6.5.3 The Bankers Algorithm for a Single Resource

6.5.4 The Bankers Algorithm for Multiple Resources

6.6 Deadlock Prevention

6.6.1 Attacking the Mutual-Exclusion Condition

6.6.2 Attacking the Hold-and-Wait Condition

6.6.3 Attacking the No-Preemption Condition

6.6.4 Attacking the Circular Wait Condition

6.7 Other Issues

6.7.1 Two-Phase Locking

6.7.2 Communication Deadlocks

6.7.3 Livelock

6.7.4 Starvation

6.8 Research On Deadlocks

6.9 Summary

Chapter 7. Virtualization And The Cloud

7.1 History

7.2 Requirements For Virtualization

7.3 Type 1 And Type 2 Hypervisors

Table of Contents

7.4 Techniques For Efficient Virtualization

7.4.1 Virtualizing the Unvirtualizable

7.4.2 The Cost of Virtualization

7.5 Are Hypervisors Microkernels Done Right?

7.6 Memory Virtualization

7.7 I/O Virtualization

7.8 Virtual Machines On Multicore CPUs

7.9 Clouds

7.9.1 Clouds as a Service

7.9.2 Virtual Machine Migration

7.9.3 Checkpointing

7.10 OS-Level Virtualization

7.11 Case Study: VMWARE

7.11.1 The Early History of VMware

7.11.2 VMware Workstation

7.11.3 Challenges in Bringing Virtualization to the x86

7.11.4 VMware Workstation: Solution Overview

7.11.5 The Evolution of VMware Workstation

7.11.6 ESX Server: VMwares type 1 Hypervisor

7.12 Research On Virtualization And The Cloud

7.13 Summary

Chapter 8. Multiple Processor Systems

8.1 Multiprocessors

8.1.1 Multiprocessor Hardware

8.1.2 Multiprocessor Operating System Types

8.1.3 Multiprocessor Synchronization

8.1.4 Multiprocessor Scheduling

8.2 Multicomputers

Table of Contents

- 8.2.1 Multicomputer Hardware
- 8.2.2 Low-Level Communication Software
- 8.2.3 User-Level Communication Software
- 8.2.4 Remote Procedure Call
- 8.2.5 Distributed Shared Memory
- 8.2.6 Multicomputer Scheduling
- 8.2.7 Load Balancing

8.3 Distributed Systems

- 8.3.1 Network Hardware
- 8.3.2 Network Services and Protocols
- 8.3.3 Document-Based Middleware
- 8.3.4 File-System-Based Middleware
- 8.3.5 Object-Based Middleware
- 8.3.6 Coordination-Based Middleware

8.4 Research On Multiple Processor Systems

8.5 Summary

Chapter 9. Security

9.1 Fundamentals Of Operating System Security

- 9.1.1 The CIA Security Triad
- 9.1.2 Security Principles
- 9.1.3 Security of the Operating System Structure
- 9.1.4 Trusted Computing Base
- 9.1.5 Attackers
- 9.1.6 Can We Build Secure Systems?

9.2 Controlling Access To Resources

- 9.2.1 Protection Domains
- 9.2.2 Access Control Lists
- 9.2.3 Capabilities

9.3 Formal Models Of Secure Systems

Table of Contents

9.3.1 Multilevel Security

9.3.2 Cryptography

9.3.3 Trusted Platform Modules

9.4 Authentication

9.4.1 Passwords

9.4.2 Authentication Using a Physical Object

9.4.3 Authentication Using Biometrics

9.5 Exploiting Software

9.5.1 Buffer Overflow Attacks

9.5.2 Format String Attacks

9.5.3 Use-After-Free Attacks

9.5.4 Type Confusion Vulnerabilities

9.5.5 Null Pointer Dereference Attacks

9.5.6 Integer Overflow Attacks

9.5.7 Command Injection Attacks

9.5.8 Time of Check to Time of Use Attacks

9.5.9 Double Fetch Vulnerability

9.6 Exploiting Hardware

9.6.1 Covert Channels

9.6.2 Side Channels

9.6.3 Transient Execution Attacks

9.7 Insider Attacks

9.7.1 Logic Bombs

9.7.2 Back Doors

9.7.3 Login Spoofing

9.8 Operating System Hardening

9.8.1 Fine-Grained Randomization

9.8.2 Control-Flow Restrictions

9.8.3 Access Restrictions

Table of Contents

9.8.4 Code and Data Integrity Checks

9.8.5 Remote Attestation Using a Trusted Platform Module

9.8.6 Encapsulating Untrusted Code

9.9 Research On Security

9.10 Summary

Chapter 10. Case Study 1: Unix, Linux, And Android

10.1 History Of Unix And Linux

10.1.1 UNICS

10.1.2 PDP-11 UNIX

10.1.3 Portable UNIX

10.1.4 Berkeley UNIX

10.1.5 Standard UNIX

10.1.6 MINIX

10.1.7 Linux

10.2 Overview Of Linux

10.2.1 Linux Goals

10.2.2 Interfaces to Linux

10.2.3 The Shell

10.2.4 Linux Utility Programs

10.2.5 Kernel Structure

10.3 Processes In Linux

10.3.1 Fundamental Concepts

10.3.2 Process-Management System Calls in Linux

10.3.3 Implementation of Processes and Threads in Linux

10.3.4 Scheduling in Linux

10.3.5 Synchronization in Linux

10.3.6 Booting Linux

10.4 Memory Management In Linux

10.4.1 Fundamental Concepts

Table of Contents

- 10.4.2 Memory Management System Calls in Linux
- 10.4.3 Implementation of Memory Management in Linux
- 10.4.4 Paging in Linux

10.5 Input/Output In Linux

- 10.5.1 Fundamental Concepts
- 10.5.2 Networking
- 10.5.3 Input/Output System Calls in Linux
- 10.5.4 Implementation of Input/Output in Linux
- 10.5.5 Modules in Linux

10.6 The Linux File System

- 10.6.1 Fundamental Concepts
- 10.6.2 File-System Calls in Linux
- 10.6.3 Implementation of the Linux File System
- 10.6.4 NFS: The Network File System

10.7 Security In Linux

- 10.7.1 Fundamental Concepts
- 10.7.2 Security System Calls in Linux
- 10.7.3 Implementation of Security in Linux

10.8 Android

- 10.8.1 Android and Google
- 10.8.2 History of Android
- 10.8.3 Design Goals
- 10.8.4 Android Architecture
- 10.8.5 Linux Extensions
- 10.8.6 ART
- 10.8.7 Binder IPC
- 10.8.8 Android Applications
- 10.8.9 Intents
- 10.8.10 Process Model

Table of Contents

10.8.11 Security and Privacy

10.8.12 Background Execution and Social Engineering

10.9 Summary

Chapter 11. Case Study 2: Windows 11

11.1 History Of Windows Through Windows 11

11.1.1 1980s: MS-DOS

11.1.2 1990s: MS-DOS-based Windows

11.1.3 2000s: NT-based Windows

11.1.4 Windows Vista

11.1.5 Windows 8

11.1.6 Windows 10

11.1.7 Windows 11

11.2 Programming Windows

11.2.1 Universal Windows Platform

11.2.2 Windows Subsystems

11.2.3 The Native NT Application Programming Interface

11.2.4 The Win32 Application Programming Interface

11.2.5 The Windows Registry

11.3 System Structure

11.3.1 Operating System Structure

11.3.2 Booting Windows

11.3.3 Implementation of the Object Manager

11.3.4 Subsystems, DLLs, and User-Mode Services

11.4 Processes And Threads In Windows

11.4.1 Fundamental Concepts

11.4.2 Job, Process, Thread, and Fiber Management API Calls

11.4.3 Implementation of Processes and Threads

11.4.4 WoW64 and Emulation

11.5 Memory Management

Table of Contents

- 11.5.1 Fundamental Concepts
- 11.5.2 Memory-Management System Calls
- 11.5.3 Implementation of Memory Management
- 11.5.4 Memory Compression
- 11.5.5 Memory Partitions

11.6 Caching In Windows

11.7 Input/Output In Windows

- 11.7.1 Fundamental Concepts
- 11.7.2 Input/Output API Calls
- 11.7.3 Implementation of I/O

11.8 The Windows NT File System

- 11.8.1 Fundamental Concepts
- 11.8.2 Implementation of the NT File System

11.9 Windows Power Management

11.10 Virtualization In Windows

- 11.10.1 Hyper-V
- 11.10.2 Containers
- 11.10.3 Virtualization-Based Security

11.11 Security In Windows

- 11.11.1 Fundamental Concepts
- 11.11.2 Security API Calls
- 11.11.3 Implementation of Security
- 11.11.4 Security Mitigations

11.12 Summary

Chapter 12. Operating System Design

12.1 The Nature Of The Design Problem

- 12.1.1 Goals
- 12.1.2 Why Is It Hard to Design an Operating System?

12.2 Interface Design

Table of Contents

- 12.2.1 Guiding Principles
- 12.2.2 Paradigms
- 12.2.3 The System-Call Interface

12.3 Implementation

- 12.3.1 System Structure
- 12.3.2 Mechanism vs. Policy
- 12.3.3 Orthogonality
- 12.3.4 Naming
- 12.3.5 Binding Time
- 12.3.6 Static vs. Dynamic Structures
- 12.3.7 Top-Down vs. Bottom-Up Implementation
- 12.3.8 Synchronous vs. Asynchronous Communication
- 12.3.9 Useful Techniques

12.4 Performance

- 12.4.1 Why Are Operating Systems Slow?
- 12.4.2 What Should Be Optimized?
- 12.4.3 Space-Time Trade-offs
- 12.4.4 Caching
- 12.4.5 Hints
- 12.4.6 Exploiting Locality
- 12.4.7 Optimize the Common Case

12.5 Project Management

- 12.5.1 The Mythical Man Month
- 12.5.2 Team Structure
- 12.5.3 The Role of Experience
- 12.5.4 No Silver Bullet

Chapter 13. Reading List And Bibliography

13.1 Suggestions For Further Reading

- 13.1.1 Introduction

Table of Contents

13.1.2 Processes and Threads

13.1.3 Memory Management

13.1.4 File Systems

13.1.5 Input/Output

13.1.6 Deadlocks

13.1.7 Virtualization and the Cloud

13.1.8 Multiple Processor Systems

13.1.9 Security

13.1.10 Case Study 1: UNIX, Linux, and Android

13.1.11 Case Study 2: Windows

13.1.12 Operating System Design

13.2 Alphabetical Bibliography

Index