

1. Record Nr.	UNINA9910153253803321
Autore	Tanenbaum Andrew S. <1944->
Titolo	Modern operating systems // Andrew S. Tanenbaum and Herbert Bos
Pubbl/distr/stampa	Harlow, England : , : Pearson, , [2014] Â©2014
ISBN	9781292061955 1-292-06195-2
Edizione	[Global edition, Fourth edition.]
Descrizione fisica	1 online resource (1,106 pages)
Collana	Always Learning
Disciplina	005.4469
Soggetti	Sistemes operatius (Ordinadors) Operating systems (Computers) Llibres electrònics
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Cover -- Title -- Content -- PREFACE -- 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 (1945-55): Vacuum Tubes -- 1.2.2 The Second Generation (1955-65): 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 Disks -- 1.3.4 I/O Devices -- 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 Multiprocessor Operating Systems -- 1.4.4 Personal Computer Operating Systems -- 1.4.5 Handheld Computer Operating Systems -- 1.4.6 Embedded Operating Systems -- 1.4.7 Sensor-Node Operating Systems -- 1.4.8 Real-Time Operating Systems -- 1.4.9 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 Win32 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 -- 1.7.5 Virtual Machines -- 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 -- 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 Scheduler Activations -- 2.2.8 Pop-Up Threads -- 2.2.9 Making Single-Threaded Code Multithreaded -- 2.3 INTERPROCESS COMMUNICATION -- 2.3.1 Race Conditions -- 2.3.2 Critical Regions -- 2.3.3 Mutual Exclusion with Busy Waiting -- 2.3.4 Sleep and Wakeup -- 2.3.5 Semaphores -- 2.3.6 Mutexes -- 2.3.7 Monitors -- 2.3.8 Message Passing -- 2.3.9 Barriers -- 2.3.10 Avoiding Locks: Read-Copy-Update -- 2.4 SCHEDULING -- 2.4.1 Introduction to Scheduling -- 2.4.2 Scheduling in Batch Systems -- 2.4.3 Scheduling in Interactive Systems -- 2.4.4 Scheduling in Real-Time Systems -- 2.4.5 Policy Versus Mechanism -- 2.4.6 Thread Scheduling -- 2.5 CLASSICAL IPC PROBLEMS -- 2.5.1 The Dining Philosophers Problem -- 2.5.2 The Readers and Writers Problem -- 2.6 RESEARCH ON PROCESSES AND THREADS -- 2.7 SUMMARY -- 3 MEMORY MANAGEMENT -- 3.1 NO 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 -- 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 Page Size -- 3.5.4 Separate Instruction and Data Spaces -- 3.5.5 Shared Pages -- 3.5.6 Shared Libraries -- 3.5.7 Mapped Files -- 3.5.8 Cleaning Policy -- 3.5.9 Virtual Memory Interface -- 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 -- 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 -- 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 -- 4.3.7 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 -- 4.5 EXAMPLE FILE SYSTEMS -- 4.5.1 The MS-DOS File System -- 4.5.2 The UNIX V7 File System -- 4.5.3 CD-ROM File Systems -- 4.6 RESEARCH ON FILE SYSTEMS -- 4.7 SUMMARY -- 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 -- 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 DISKS -- 5.4.1 Disk Hardware -- 5.4.2 Disk Formatting -- 5.4.3 Disk Arm Scheduling Algorithms -- 5.4.4 Error Handling -- 5.4.5 Stable Storage -- 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 -- 6 DEADLOCKS -- 6.1 RESOURCES -- 6.1.1 Preemptable and Nonpreemptable Resources -- 6.1.2 Resource Acquisition -- 6.2 INTRODUCTION TO DEADLOCKS -- 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 Banker's Algorithm for a Single Resource -- 6.5.4 The Banker's 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 -- 7 VIRTUALIZATION AND THE CLOUD -- 7.1 HISTORY -- 7.2 REQUIREMENTS FOR VIRTUALIZATION -- 7.3 TYPE 1 AND TYPE 2 HYPERVISORS -- 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 APPLIANCES -- 7.9 VIRTUAL MACHINES ON MULTICORE CPUS -- 7.10 LICENSING ISSUES -- 7.11 CLOUDS -- 7.11.1 Clouds as a Service -- 7.11.2 Virtual Machine Migration -- 7.11.3 Checkpointing -- 7.12 CASE STUDY: VMWARE -- 7.12.1 The Early History of VMware -- 7.12.2 VMware Workstation -- 7.12.3 Challenges in Bringing Virtualization to

the x86 -- 7.12.4 VMware Workstation: Solution Overview -- 7.12.5
The Evolution of VMware Workstation -- 7.12.6 ESX Server: VMware's
type 1 Hypervisor -- 7.13 RESEARCH ON VIRTUALIZATION AND THE
CLOUD -- 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 -- 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 -- 8.3 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.

Sommario/riassunto

Modern Operating Systems, Fourth Edition, is intended for introductory courses in Operating Systems in Computer Science, Computer Engineering, and Electrical Engineering programs. The widely anticipated revision of this worldwide best-seller incorporates the latest developments in operating systems (OS) technologies. The Fourth Edition includes up-to-date materials on relevant OS. Tanenbaum also provides information on current research based on his experience as an operating systems researcher. Modern Operating Systems, Third Edition was the recipient of the 2010 McGuffey Longevity Award. The McGuffey Longevity Award recognizes textbooks whose excellence has been demonstrated over time. <http://taaonline.net/index.html>

Teaching and Learning Experience This program will provide a better teaching and learning experience-for you and your students. It will help:

- Provide Practical Detail on the Big Picture Concepts: A clear and entertaining writing style outlines the concepts every OS designer needs to master.
- Keep Your Course Current: This edition includes information on the latest OS technologies and developments
- Enhance Learning with Student and Instructor Resources: Students will gain hands-on experience using the simulation exercises and lab experiments.
