An Introduction to Operating Systems

Revised First Edition

Harvey M. Deitel

Including Case Studies in:

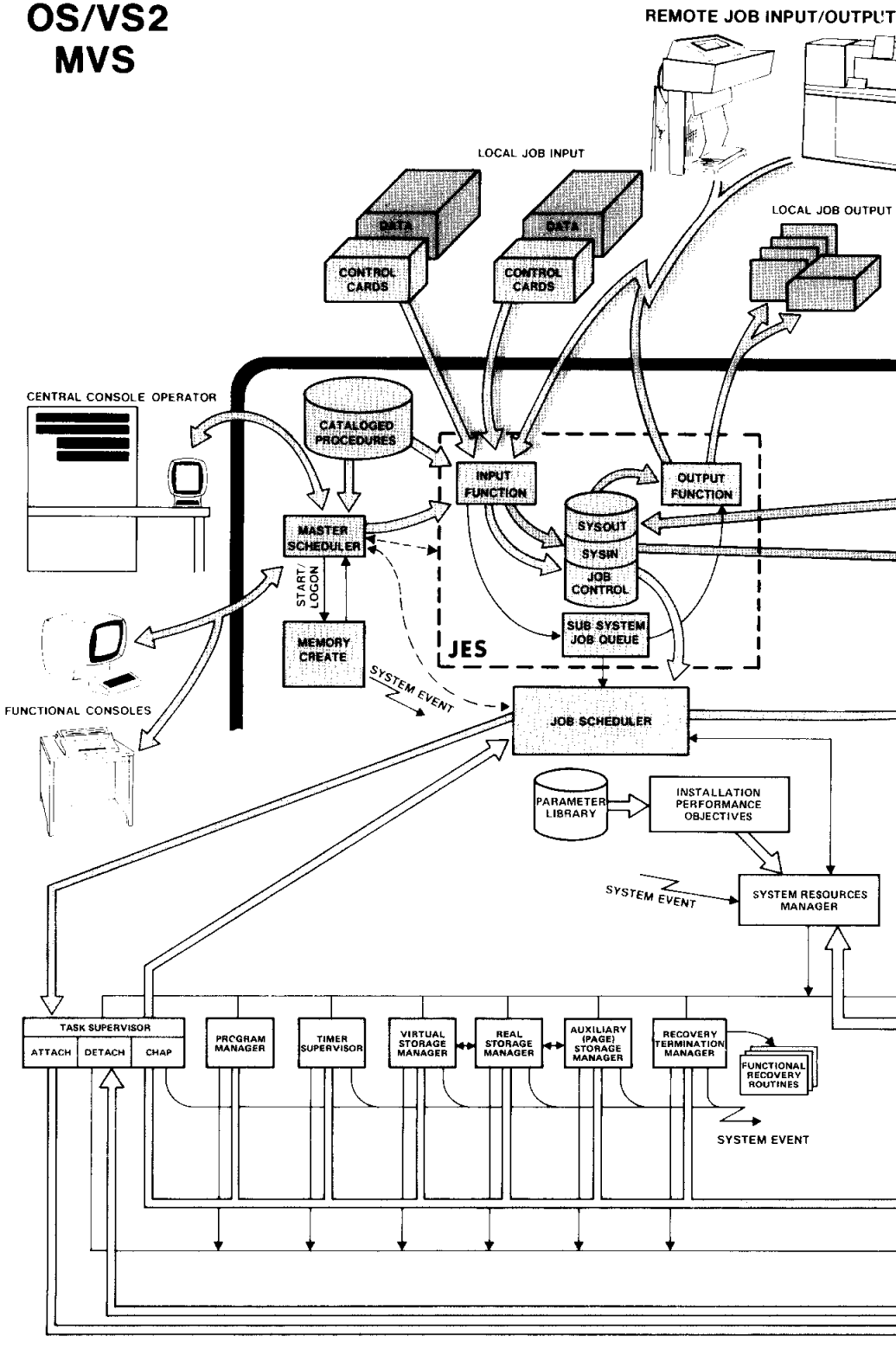
UNIX™
VAX™
CP/M®
MVS
VM

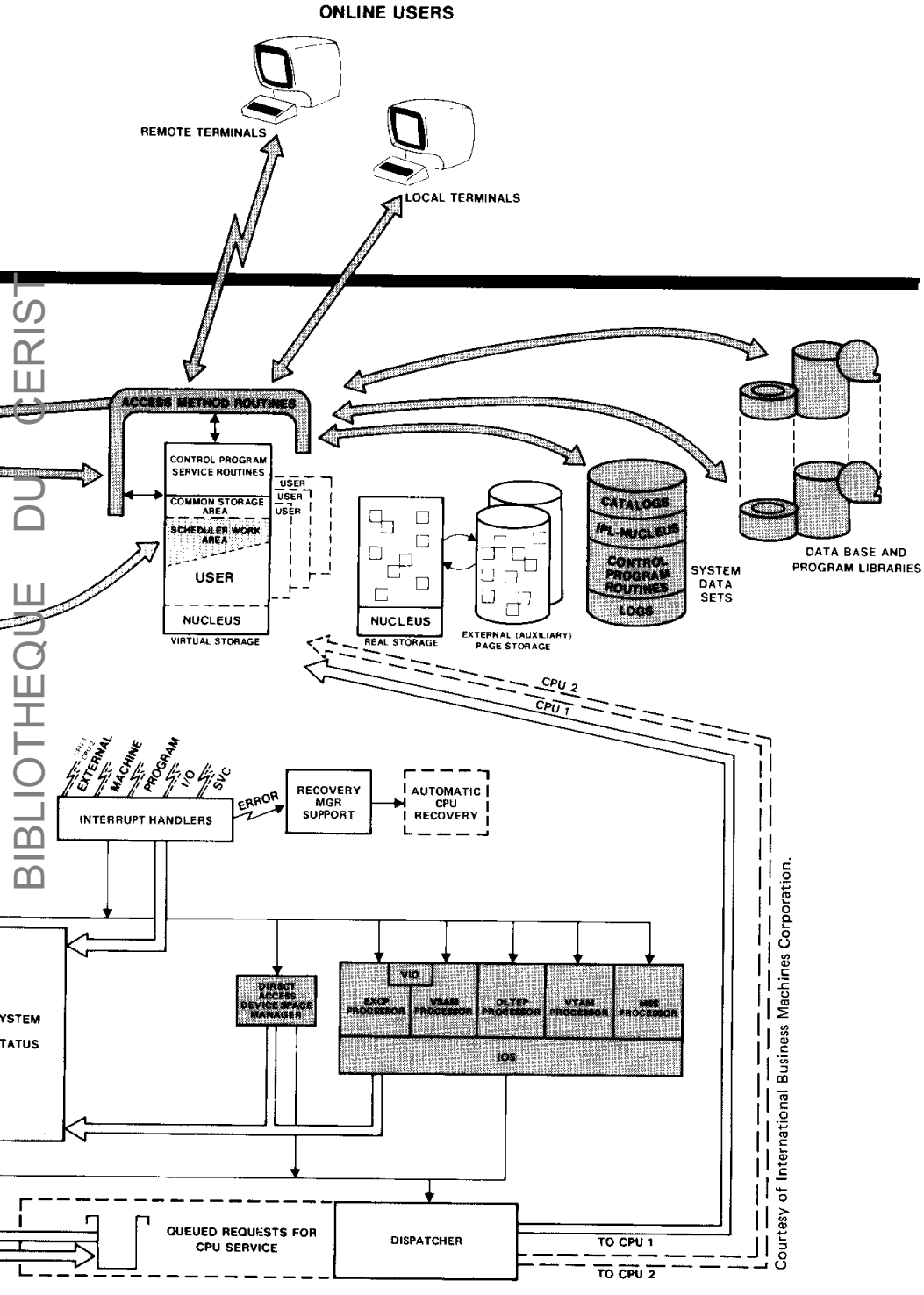
With Concurrent Programming in Ada®

An Introduction to Operating Systems

Revised First Edition

OS/VS2 MVS





An Introduction to Operating Systems

Revised First Edition

HARVEY M. DEITEL
Boston College

ADDISON-WESLEY PUBLISHING COMPANY
Reading, Massachusetts • Menlo Park, California
Don Mills, Ontario • Wokingham, England • Amsterdam • Bonn • Sydney
Singapore • Tokyo • Madrid • Bogota • Santiago • San Juan

WORLD STUDENT SERIES

Sponsoring Editor: William B. Gruener
Production Editor: William J. Yskamp

Text Designer: Herb Caswell
Illustrator: ANCO/Boston
Cover Designer: Gary Fujiwara
Art Coordinator: Susanah H. Michener

Production Manager: Susan Zorn
Production Coordinator: Helen M. Wythe

The text of this book was composed in
Times Roman by International Computaprint Corporation

ISBN 0-201-14502-2
EFGHIJ-HA-89876

To my wife, Barbara,
and to my children, Paul and Abbey:
You are the joys of my life.

Preface

This text is intended primarily for the one-semester operating systems course that universities offer to juniors, seniors, and first-year graduate students in computer science. It contains sufficient material to be useful as the main text for courses CS6 and CS10 in ACM Curriculum 78. Chapter 15, Analytic Modeling, is heavily mathematical; this chapter should be included by instructors in "hard-core" computer science schools, but it may be omitted without loss of continuity in less formal programs.

The text contains approximately 300 charts, diagrams, and illustrations, 504 exercises, 22 chapters, and five detailed case studies. Each chapter lists extensive literature—577 books and papers are referenced. Over 1200 terms are highlighted in the end-of-chapter terminology sections.

Five detailed case studies are included on operating systems that will have great importance in the 1980s, namely, the UNIX system, VAX/VMS, CP/M, MVS, and VM. Each of these systems has a certain "flavor" that I have worked hard to convey. Each of them is intended for a different portion of the operating systems marketplace.

Eight major parts are included; each part contains several related chapters. The text is handsomely illustrated (thanks to the skills of Sue Michener of Addison-Wesley). Each chapter begins with an outline so that the reader may approach the material in "top-down" fashion. One or more quotes are used in introducing each chapter—some are humorous; some are thought-provoking; all are intended to humanize the text and add a touch of the philosophical.

Significant terms, major concepts, and important statements are italicized. Each chapter includes a summary of important concepts and ideas for review. Terminology sections are included; key terms are presented alphabetically. Each chapter includes many exercises varying in difficulty from simple review of the material to

complex reasoning from basic principles. Literature sections are localized; each chapter ends with a listing of relevant texts and papers. An Author Index at the end of the text lists all of the authors and co-authors mentioned in the text; an extensive conventional Subject Index provides rapid access to virtually any portion of the text by keyword.

Part One introduces the notion of operating systems, presents a history of operating systems, and discusses hardware, software, and firmware. Chapter 2 includes a case study on microprogramming; this is especially important to the operating systems student—in future designs much (and in some cases all) of the operating system will migrate into microcode. Portions of Chapter 1 are based upon "A History of Operating Systems" by N. Weizer, *Datamation*, January 1981. This material is reprinted with permission of *Datamation*® magazine, © Copyright by Technical Publishing Company, A Dun & Bradstreet Company, 1981, all rights reserved.

Part Two presents the notions of process, process state transitions, interrupts, context switching, operating systems structure, asynchronism, mutual exclusion, monitors, and deadlock. Chapter 3 introduces various process concepts and discusses the interrupt structure of the large-scale IBM processors; this material is critical to the discussions of the MVS and VM operating systems in the case studies later in the text. Chapter 4 presents the notion of asynchronism. The problems encountered in concurrent access to shared resources are discussed, and various mutual exclusion techniques, both hardware- and software-oriented, are presented for dealing with these problems. Chapter 5 discusses monitors and how they may be used to solve certain classical problems in concurrency; monitor implementations of the Ring Buffer as well as Readers and Writers are presented. The chapter continues with an introduction to concurrent programming in Ada, the emerging language whose development has been sponsored by the United States government.

Many of the Ada program segments presented are based on examples in *Preliminary Ada Reference Manual*, SIGPLAN NOTICES, Vol. 14, No. 6, June 1979, Part A, and *Rationale for the Design of the Ada Programming Language*, by J. D. Ichbiah, J. C. Heliard, O. Roubine, J. G. P. Barnes, B. Krieg-Brueckner, and B. A. Wichmann, SIGPLAN NOTICES, Vol. 14, No. 6, June 1979, Part B ("The Government of the United States of America grants free permission to reproduce this document for the purpose of evaluating and using the Ada language"). (Ada® is a registered trademark of the United States Government, Ada Joint Program Office.)

Chapter 6 explains the notion of deadlock in which various processes cannot proceed because they are waiting for events that will never happen. The chapter discusses the major areas of deadlock research, and presents various means of dealing with deadlock and the related problem of indefinite postponement. The chapter includes a detailed discussion of deadlock avoidance with Dijkstra's Banker's Algorithm.

Part Three discusses storage management for both real storage and virtual storage systems. Chapter 7 traces the development of real storage systems from single user dedicated systems through the various forms of partitioned multiprogram-

ming systems. Chapters 8 and 9 deal with virtual storage organization and management, respectively. Chapter 8 motivates the concept of virtual storage, and discusses the reduction of mapping information by the block mapping techniques and paging and segmentation. A detailed discussion of virtual storage organization and address translation in paged/segmented systems is presented. Chapter 9 discusses the various strategies for managing virtual storage systems. Fetch, placement, and replacement strategies are considered; both demand fetch and anticipatory fetch strategies are discussed. The chapter concentrates on page replacement strategies; it considers the Principle of Optimality, and the random, FIFO, LRU, LFU, and NUR strategies. Then, Denning's working set theory of program behavior is presented, and working set page replacement is analyzed.

Part Four deals with processor management, in particular the issues of processor scheduling and multiprocessing. Chapter 10 concentrates on scheduling strategies; it discusses high-level, intermediate-level, and low-level scheduling, scheduling objectives and criteria, priority scheduling, static vs. dynamic priorities, earned vs. bought priorities, and deadline scheduling. Various scheduling algorithms are presented including FIFO, RR, SJF, SRT, and HRN. The chapter ends with a thorough analysis of multilevel feedback queueing mechanisms. Chapter 11 explains multiprocessing, motivated as a means of improving performance and reliability. The exploitation of parallelism is considered; the techniques of loop distribution, tree height reduction, and the "never wait" rule are discussed. Various multiprocessor hardware organizations and operating systems organizations are analyzed. This material has strong ties to the material on networks in Chapter 16.

Part Five considers auxiliary storage management. Chapter 12 discusses disk scheduling; it explains the operation of moving-head disk storage, motivates the need for disk scheduling, and presents the popular disk scheduling strategies including FCFS, SSTF, SCAN, N-step SCAN, C-SCAN, the Eschenbach scheme and rotational optimization. A number of systems considerations that might affect the usefulness of disk scheduling are examined.

Chapter 13 considers file and database management systems. The chapter discusses file system functions and operations; the data hierarchy; blocking and buffering; sequential, indexed sequential, direct, and partitioned file organizations; queued and basic access methods; hierarchical file system structure; contiguous and noncontiguous allocation; linked allocation, file mapping, file descriptors, access control; and backup and recovery. The importance of database systems, especially in the context of operating systems, is emphasized; the discussion considers the advantages of database systems, data independence, database languages, the database administrator, distributed database, and data dictionary concepts. The chapter ends with a discussion of the hierarchical, network, and relational database models.

Part Six deals with the issues of computer system performance. Chapter 14 considers the issues of performance measurement, monitoring, and evaluation. The chapter discusses timings, instruction mixes, kernel programs, analytic models, benchmarks, synthetic programs, simulation, and performance monitoring. Bottle-

neck isolation and removal are examined. Both negative and positive feedback mechanisms are analyzed. The chapter includes a very large complement of exercises, many of which are suitable as term projects, particularly in simulation-oriented courses.

Chapter 15 presents a mathematical treatment of analytic modeling; both queueing models and Markov processes are considered. The chapter may be omitted without loss of continuity, but it is highly recommended for students who have had some background in calculus, probability, and statistics. The queueing theory portion of the chapter discusses the notions of source, arrivals, Poisson arrivals, service times, queue capacity, multiple servers, queue disciplines, traffic intensity, server utilization, steady state vs. transient solutions, and Little's Result. Two case studies are presented: analyzing an M/M/1 queueing system, and analyzing an M/M/c queueing system. The section of the chapter on Markov processes concentrates on the special case of birth and death models; a rather detailed case study analyzes the performance of a disk subsystem. Portions of the presentation on queueing theory are based on material in Chapter 5 of *Probability, Statistics, and Queueing Theory with Computer Science Applications* by Arnold O. Allen. Copyright 1978 by Academic Press, Inc., New York. Adapted with permission.

Part Seven considers computer networks and security issues. Chapter 16, Network Operating Systems, discusses network categories including resource sharing networks, distributed computation networks, and remote communication networks. A detailed discussion of packet switching is presented; included is the ISO OSI layered architecture, the X.25 standard, and the datagram and virtual circuit models. Network operating systems are examined as well as the primitive operations required to service them—user communication, job migration, data migration, and control primitives. The popular network organizations are examined including the star, ring, and mesh topologies. Notions of security, privacy, encryption, and authentication are discussed; this is preliminary to the more detailed presentation in Chapter 17. Local networking is considered in depth; the techniques of CSMA/CD, token passing, and message slots are explained. The operation of Ethernet is discussed. The chapter ends with a case study of Digital Equipment Corporation's DECnet and its DNA (Digital Network Architecture). This case study emphasizes the functions and capabilities provided in a major network operating system, especially those of file handling and interprocess communication.

Chapter 17 deals with computer security and its importance in operating systems. The chapter discusses security requirements, the need for a total approach to security, external security, operational security, surveillance, threat monitoring, amplification, password protection, auditing, access controls, security kernels, hardware security, and survivable systems. It continues with a detailed discussion of the capabilities-based, object-oriented systems that are receiving so much attention today as the means for achieving systems that afford greater security. A case study is presented on the object-oriented architecture of the IBM System/38. Cryptography is discussed; a cryptographic privacy system is illustrated; cryptanalysis, public key systems, digital signatures, and the DES and RSA schemes are explained; numerous applications of cryptography are listed. The DES standards document published by

the United States Government is included as an appendix to the text. Operating system penetration is considered in depth; generic system functional flaws and generic operating system attacks are categorized. The chapter concludes with a case study examining the successful penetration of an existing operating system.

Part Eight presents five detailed case studies on important operating systems, namely, the UNIX system, VAX/VMS, CP/M, MVS, and VM. The body of the text discusses general principles; the case studies deal with the real issues of real operating systems.

Chapter 18 discusses the UNIX operating system developed by Bell Laboratories. The UNIX system has been nothing less than a phenomenon of its time. It was developed primarily by two people for their own use. Now it is used in universities and industrial environments throughout the world. Versions developed for personal computers are becoming quite popular; some enthusiasts believe the UNIX system will eventually displace CP/M as the de facto standard personal computer operating system. (This issue is investigated further in Chapter 20.) The case study includes a discussion of the interesting UNIX input/output system, and the UNIX-originated concepts of pipes and filters. Within the UNIX system case study is another case study on Microsoft's XENIX system—a version of the UNIX system developed for 16-bit microcomputers. Microsoft supplies a XENIX-like system for the IBM Personal Computer.

Chapter 19 discusses Digital Equipment Corporation's top-of-the-line operating system, VAX/VMS. VAX extends the 16-bit architecture of DEC's successful PDP-11 series into the realm of today's popular 32-bit virtual storage systems. The case study focuses on storage management, process scheduling, input/output, record management services, interprocess communication, and process synchronization. VAX is quite typical of today's efforts by minicomputer manufacturers to produce systems that challenge the power of large mainframe systems. Major portions of the VAX case study are based upon *Computer Programming and Architecture: The VAX-11* by H. M. Levy and R. H. Eckhouse, Jr., Digital Press, 1980, and *VAX-11 Software Handbook*, Digital Equipment Corporation, 1981. The author is grateful for permission granted to use these materials.

Chapter 20 discusses Digital Research's CP/M operating system for microcomputers. Initially developed for the 8-bit micros that served at the heart of early personal computers like Radio Shack's TRS-80 Model I, CP/M has been extended considerably to compete in the 16-bit arena—it is offered as CP/M-86 for the 16-bit IBM Personal Computer. The case study discusses PL/M, portability, table-driven systems, memory allocation, logical-to-physical device mapping, the Console Command Processor (CCP), the Basic Input/Output System (BIOS), the Basic Disk Operating System (BDOS), and the file system. The chapter includes two case-studies-within-case-studies: one on MP/M, a multiuser CP/M-based timesharing system, and one on CP/NET, a networking system for CP/M-based and MP/M-based microcomputers.

Chapter 21 discusses IBM's top-of-the-line operating system, MVS, designed for very large-scale processors. The chapter traces the history of IBM operating systems development since the announcement of the System/360. Important aspects of IBM

hardware architecture are reviewed. Then, many aspects of MVS are examined in detail—MVS functions, the Supervisor, the Master Scheduler, the Job Entry Subsystem, the System Management Facility, the System Activity Measurement Facility, the Timesharing Option, data management, the Real Storage Manager, the Auxiliary Storage Manager, the Virtual Storage Manager, the System Resources Manager, storage organization, resource control, locking, Enqueue, Reserve, tasks, service requests, multiprocessing, performance, and monitoring system activity.

Chapter 22 discusses what is perhaps the most "exotic" of the operating systems in the case studies, namely IBM's VM virtual machine operating system. It enables one computer system to execute several operating systems simultaneously. This capability allows an installation to run dramatically different operating systems at once, or to run different versions of the same system—perhaps allowing a new release to be tested while a production system continues operation. VM has particularly interesting ramifications in networking, as is evidenced by IBM's Hydra concept and Spartacus Computers' Kangaroo (*Datamation*, August 1981). The chapter discusses the history of VM, the Control Program (CP), demand paging, minidisks, console management, user privilege classes, the VM directory, the Conversational Monitor System (CMS), the Remote Spooling and Communications System, performance considerations, the Virtual Machine Assist Feature, the Extended Control Program Support Feature, performance measurement, performance analysis, reliability, availability, and serviceability. The case study ends with a discussion of why VM may emerge as IBM's banner operating system for large-scale processors for the mid-to-late 1980s.

The richness of the case studies provides the material for comparing and contrasting the different design and implementation philosophies used in contemporary operating systems. The case studies span the full range from "tiny" operating systems like CP/M to massive systems like MVS. They include mainframe operating systems like MVS and VM, minicomputer operating systems like VAX/VMS and UNIX, and microcomputer operating systems like CP/M and XENIX. They include real storage systems and virtual storage systems, real machine systems and virtual machine systems, single user systems and multiuser systems, stand-alone systems, multiprocessing systems, and networked systems.

It is a pleasure to acknowledge the many people who have contributed to this project. The most important acknowledgment is to the hundreds of authors represented in the literature sections in the chapters; their fine papers and texts have provided the diversity of interesting material that makes operating systems such a fascinating area. The book was reviewed by experts in many fields. The core of the review team consisted of James Peterson of the University of Texas at Austin, Richard Wexelblatt of Sperry Univac, Paul Ross of Millersville State College, and Anthony Lucido of Intercomp.

The sections of the book dealing with computer hardware, microprogramming, and computer security were reviewed by Steve Paris of Prime Computer. Bart Guerreri, president of DSD Laboratories, contributed considerable hardware exper-

tise. Nathan Tobol, consulting engineer for Codex Corporation and chairman of the IEEE 802 Local-Area Networking subcommittee, reviewed Chapter 16, Network Operating Systems, and contributed valuable material from one of his forthcoming papers. Larry Nelson, Chief of Systems Support at AVCO Services, contributed most of the material in the MVS case study, and provided many useful insights into VM. Barry Shein, who administered UNIX and VAX systems for Harvard University, contributed many useful suggestions that helped me refine those case studies. William Blocher, president of The Bit Bucket, provided insights on CP/M.

Eliezer Gafni of MIT, and Anat Gafni and Josefina Bondoc of Boston University scrupulously examined Chapter 15, on Analytic Modeling; the material on queueing theory and Markov processes was polished considerably by the incorporation of their numerous suggestions.

Others who provided assistance and inspiration throughout the writing effort were Julius Zigman, Miriam Zigman, Stephanie Guerreri, Eric Shoemaker, Mary Wiles, Steven Broderick, Richard Miles, and Shirley Spas.

My production editor, William Yskamp of Addison-Wesley, has done an impeccable job bringing this book to publication. His dedication to the project never wavered, his influence on the form and style of the finished product was truly substantial, and his leadership was critical to the timely publication of the text.

My wife, Barbara, and my children, Paul and Abbey, provided the incredible support and understanding without which this text could never have come to fruition. They collectively contributed 2000 hours of researching, sorting, copying, word processing, and proofreading. Their efforts helped me trim one year from the writing schedule for the text.

William B. Gruener, Executive Editor, the computer sciences, Addison-Wesley Publishing Company, has been my mentor in publishing since 1978. Bill is truly responsible for the fact that this book happened. He and Addison-Wesley placed impressive resources at my disposal. I hope this work merits the trust and confidence they have shown in me.

Any effort of this scope is bound to have its flaws. I assume complete responsibility for any remaining defects. I am currently researching and writing the next edition of this work, and would be most grateful for your comments, criticisms, and corrections. Any correspondence should be sent to Harvey M. Deitel (author), c/o Computer Science Editor, Addison-Wesley Publishing Company, Reading, Mass. 01867. I will acknowledge all correspondence immediately.

H. M. D.

Framingham, Massachusetts
August 1983

Contents

PART 7
NETWORKS AND SECURITY