

# THE LOGICAL DESIGN OF OPERATING SYSTEMS

**ALAN C. SHAW**

*Computer Science Group  
University of Washington*

Technische Hochschule Darmstadt  
FACHBEREICH INFORMATIK  
B I B L I O T H E K  
Inventar-Nr.: 2619  
Sachgebiete: \_\_\_\_\_  
Standort: \_\_\_\_\_

**PRENTICE-HALL, INC.**

ENGLEWOOD CLIFFS, N. J.

# CONTENTS

PREFACE      xiii

## **1** THE ORGANIZATION OF COMPUTING SYSTEMS      1

- 1.1 Some Definitions      1
- 1.2 Notation for Algorithms      3
- 1.3 Historical Perspective      5
  - 1.3.1 *Early Systems*      5
  - 1.3.2 *Second Generation Hardware and Software*      8
  - 1.3.3 *Systems of the Third Generation and Beyond*      9
- 1.4 Some Views of Operating Systems      10
  - 1.4.1 *Virtual Machines, Translation, and Resource Allocation*      11
  - 1.4.2 *Four Key Issues*      14
- 1.5 Systems Organization      14

## **2** BATCH PROCESSING SYSTEMS      17

- 2.1 Introduction      17
- 2.2 Linking and Loading      18
  - 2.2.1 *Static Relocation*      20
  - 2.2.2 *The Linking Process*      20
- 2.3 Input-Output Methods      25
  - 2.3.1 *Direct IO*      27
  - 2.3.2 *Indirect IO*      28
- 2.4 IO Buffering      29
  - 2.4.1 *CPU Interrogates Channel*      30
  - 2.4.2 *Multiple Buffers and a Coroutine Program Structure*      32
  - 2.4.3 *Channel Interrupts CPU*      38
  - 2.4.4 *Buffer Pooling for Input and Output*      41
- 2.5 The IO Supervisor      49

<b>3</b>	<b>INTERACTING PROCESSES</b>	<b>50</b>
3.1	Parallel Programming	50
3.1.1	<i>Applications</i>	50
3.1.2	<i>Some Programming Constructs for Parallelism</i>	54
3.2	The Concept of a Process	58
3.3	The Critical Section Problem	59
3.3.1	<i>The Problem</i>	59
3.3.2	<i>Software Solution (Dijkstra, 1965a, 1968b)</i>	61
3.4	Semaphore Primitives	65
3.4.1	<i>The P and V Operations</i>	65
3.4.2	<i>Mutual Exclusion Using Semaphore Operations</i>	66
3.4.3	<i>Semaphores as Resource Counters and Synchronizers in Producer Consumer Problems</i>	67
3.5	Implementing Semaphore Operations	80
3.5.1	<i>Implementation With a Busy Wait</i>	81
3.5.2	<i>Avoiding the Busy Wait</i>	82
3.6	Other Synchronization Primitives	84
<b>4</b>	<b>INTRODUCTION TO MULTIPROGRAMMING SYSTEMS</b>	<b>88</b>
4.1	Rationale for Multiprogramming	88
4.2	Systems Components	90
4.2.1	<i>Hardware Features</i>	90
4.2.2	<i>Basic Software</i>	92
4.3	Operating System Nucleus	96
4.4	The User Interface	102
4.4.1	<i>Command and Control Languages</i>	102
4.4.2	<i>Job Control</i>	103
4.5	Elements of a Design Methodology	106
<b>5</b>	<b>MAIN STORAGE MANAGEMENT</b>	<b>109</b>
5.1	Static and Dynamic Relocation	109
5.1.1	<i>Hardware Address Relocation</i>	109
5.1.2	<i>Arguments for Static and Dynamic Relocation</i>	111
5.1.3	<i>Virtual Memories</i>	114
5.2	Principles of Segmentation and Paging	116
5.2.1	<i>Single Segment Name Space</i>	116
5.2.1.1	<i>Contiguous Storage Allocation</i>	116
5.2.1.2	<i>Paging</i>	117
5.2.2	<i>Multiple Segment Name Space</i>	122
5.2.2.1	<i>Contiguous Allocation Per Segment</i>	122
5.2.2.2	<i>Paging With Segmentation</i>	123

5.3	Real and Virtual Storage Protection	124
5.4	Allocation Strategies	130
5.4.1	<i>Storage Allocation in Non-Paged Systems</i>	130
5.4.2	<i>Allocation in Paged Systems</i>	138
5.4.2.1	<i>Static and Dynamic Allocation</i>	138
5.4.2.2	<i>Replacement Schemes</i>	141
5.5	Evaluation of Paging	144
5.6	Storage Hierarchies	148

## **6 PROCEDURE AND DATA SHARING IN MAIN STORAGE 152**

6.1	Why Share?	152
6.2	Requirements for Code Sharing	153
6.3	Sharing in Statically Allocated Systems	156
6.4	Dynamic Sharing	158
6.4.1	<i>Form of a Procedure Segment</i>	159
6.4.2	<i>Data Linkage</i>	161
6.4.3	<i>Procedure References</i>	163

## **7 PROCESS AND RESOURCE CONTROL 166**

7.1	Data Structures for Processes and Resources	167
7.1.1	<i>Process Descriptors</i>	167
7.1.2	<i>Resource Descriptors</i>	171
7.2	Basic Operations on Processes and Resources	177
7.2.1	<i>Process Control</i>	177
7.2.2	<i>Resource Primitives</i>	181
7.2.3	<i>Capabilities</i>	186
7.3	Interrupts and Input-Output Processes	187
7.4	Organization of Process Schedulers	191
7.4.1	<i>Master and Shared Schedulers</i>	193
7.4.2	<i>Priority Scheduling</i>	194
7.5	Scheduling Methods	198

## **8 THE DEADLOCK PROBLEM 203**

8.1	Examples of Deadlock in Computer Systems	204
8.2	A Systems Model	209
8.3	Deadlock With Serially-Reusable Resources	213
8.3.1	<i>Reusable Resource Graphs</i>	213
8.3.2	<i>Deadlock Detection</i>	215
8.3.3	<i>Recovery from Deadlock</i>	224
8.3.4	<i>Prevention Methods</i>	227
8.4	Consumable Resource Systems	232
8.5	General Resource Graphs	239
8.6	Dynamic Addition and Removal of Processes and Resources	240

# 9

## FILE SYSTEMS 244

9.1	Virtual and Real File Storage	245
9.2	File System Components	247
9.3	Logical and Physical Organizations	252
9.4	Accessing Procedures	256
9.4.1	<i>File Directories</i>	256
9.4.2	<i>Descriptors for Files</i>	260
9.4.3	<i>Access Control</i>	261
9.4.4	<i>Opening and Closing Routines</i>	264
9.5	Management of Auxiliary Storage Space	265
9.6	A Hierarchical Model for File Systems	268
9.7	Recovery from Systems Failure	272

## Appendix: A MULTIPROGRAMMING PROJECT 277

A1.	Introduction	277
A2.	Machine Specifications	278
A3.	Job, Program, and Data Card Formats	283
A4.	The Operating System	284
A5.	Project Requirements	285
A6.	Some Limitations	286

## REFERENCES 287

## INDEX 293