Contents

List of Figures **vi**

Preface **xix**

1 Introduction **1**
1.1 Introduction . . . 1
1.2 Distributed Systems versus Parallel Systems . . . 3
1.3 Overview of the Book . . . 4
1.4 Characteristics of Parallel and Distributed Systems . . . 6
1.5 Design Goals . . . 7
1.6 Specification of Processes and Tasks . . . 8
1.6.1 Runnable Interface . . . 11
1.6.2 Join Construct in Java . . . 11
1.6.3 Thread Scheduling . . . 13
1.7 Problems . . . 13
1.8 Bibliographic Remarks . . . 15

2 Mutual Exclusion Problem **17**
2.1 Introduction . . . 17
2.2 Peterson's Algorithm . . . 20
2.3 Lamport's Bakery Algorithm . . . 24
2.4 Hardware Solutions . . . 27
2.4.1 Disabling Interrupts . . . 27
2.4.2 Instructions with Higher Atomicity . . . 27
2.5 Problems . . . 28
2.6 Bibliographic Remarks . . . 30

3 Synchronization Primitives **31**
3.1 Introduction . . . 31
3.2 Semaphores . . . 31

3.2.1 The Producer-Consumer Problem 33
3.2.2 The Reader-Writer Problem 36
3.2.3 The Dining Philosopher Problem 36
3.3 Monitors 42
3.4 Other Examples 46
3.5 Dangers of Deadlocks 49
3.6 Problems 50
3.7 Bibliographic Remarks 51

4 Consistency Conditions **53**
4.1 Introduction 53
4.2 System Model 54
4.3 Sequential Consistency 55
4.4 Linearizability 57
4.5 Other Consistency Conditions 60
4.6 Problems 62
4.7 Bibliographic Remarks 63

5 Wait-Free Synchronization **65**
5.1 Introduction 65
5.2 Safe, Regular, and Atomic Registers 66
5.3 Regular SRSW Register 70
5.4 SRSW Multivalued Register 71
5.5 MRSW Register 73
5.6 MRMW Register 74
5.7 Atomic Snapshots 76
5.8 Consensus 78
5.9 Universal Constructions 84
5.10 Problems 87
5.11 Bibliographic Remarks 87

6 Distributed Programming **89**
6.1 Introduction 89
6.2 InetAddress Class 89
6.3 Sockets based on UDP 90
6.3.1 Datagram Sockets 90
6.3.2 DatagramPacket Class 91
6.3.3 Example Using Datagrams 92
6.4 Sockets Based on TCP 94
6.4.1 Server Sockets 96

6.4.2 Example 1: A Name Server 96
6.4.3 Example 2: A Linker 100
6.5 Remote Method Invocations 101
6.5.1 Remote Objects 105
6.5.2 Parameter Passing 107
6.5.3 Dealing with Failures 108
6.5.4 Client Program 108
6.6 Other Useful Classes 109
6.7 Problems 109
6.8 Bibliographic Remarks 110

7 Models and Clocks 111
7.1 Introduction 111
7.2 Model of a Distributed System 112
7.3 Model of a Distributed Computation 114
7.3.1 Interleaving Model 114
7.3.2 Happened-Before Model 114
7.4 Logical Clocks 115
7.5 Vector Clocks 117
7.6 Direct-Dependency Clocks 122
7.7 Matrix Clocks 125
7.8 Problems 126
7.9 Bibliographic Remarks 127

8 Resource Allocation 129
8.1 Introduction 129
8.2 Specification of the Mutual Exclusion Problem 130
8.3 Centralized Algorithm 132
8.4 Lamport's Algorithm 135
8.5 Ricart and Agrawala's Algorithm 136
8.6 Dining Philosopher Algorithm 138
8.7 Token-Based Algorithms 142
8.8 Quorum-Based Algorithms 144
8.9 Problems 146
8.10 Bibliographic Remarks 147

9 Global Snapshot 149
9.1 Introduction 149
9.2 Chandy and Lamport's Global Snapshot Algorithm 151
9.3 Global Snapshots for non-FIFO Channels 154

9.4 Channel Recording by the Sender 154
9.5 Application: Checkpointing a Distributed Application 157
9.6 Problems 161
9.7 Bibliographic Remarks 162

10 Global Properties 163
10.1 Introduction 163
10.2 Unstable Predicate Detection 164
10.3 Application: Distributed Debugging 169
10.4 A Token-Based Algorithm for Detecting Predicates 169
10.5 Problems 173
10.6 Bibliographic Remarks 176

11 Detecting Termination and Deadlocks 177
11.1 Introduction 177
11.2 Diffusing Computation 177
11.3 Dijkstra and Scholten's Algorithm 180
11.3.1 An Optimization 181
11.4 Termination Detection without Acknowledgment Messages 182
11.5 Locally Stable Predicates 185
11.6 Application: Deadlock Detection 188
11.7 Problems 189
11.8 Bibliographic Remarks 189

12 Message Ordering 191
12.1 Introduction 191
12.2 Causal Ordering 193
12.2.1 Application: Causal Chat 196
12.3 Synchronous Ordering 196
12.4 Total Order for Multicast Messages 203
12.4.1 Centralized Algorithm 203
12.4.2 Lamport's Algorithm for Total Order 204
12.4.3 Skeen's Algorithm 204
12.4.4 Application: Replicated State Machines 205
12.5 Problems 205
12.6 Bibliographic Remarks 207

13 Leader Election 209
13.1 Introduction 209
13.2 Ring-Based Algorithms 210

13.2.1 Chang–Roberts Algorithm 210
13.2.2 Hirschberg–Sinclair Algorithm 212
13.3 Election on General Graphs 213
13.3.1 Spanning Tree Construction 213
13.4 Application: Computing Global Functions 215
13.5 Problems 217
13.6 Bibliographic Remarks 219

14 Synchronizers 221
14.1 Introduction 221
14.2 A Simple Synchronizer 223
14.2.1 Application: BFS Tree Construction 225
14.3 Synchronizer α 226
14.4 Synchronizer β 228
14.5 Synchronizer γ 230
14.6 Problems 232
14.7 Bibliographic Remarks 232

15 Agreement 233
15.1 Introduction 233
15.2 Consensus in Asynchronous Systems (Impossibility) 234
15.3 Application: Terminating Reliable Broadcast 238
15.4 Consensus in Synchronous Systems 239
15.4.1 Consensus under Crash Failures 240
15.4.2 Consensus under Byzantine Faults 243
15.5 Knowledge and Common Knowledge 244
15.6 Application: Two-General Problem 248
15.7 Problems 249
15.8 Bibliographic Remarks 250

16 Transactions 253
16.1 Introduction 253
16.2 ACID Properties 254
16.3 Concurrency Control 255
16.4 Dealing with Failures 256
16.5 Distributed Commit 257
16.6 Problems 261
16.7 Bibliographic Remarks 262

17 Recovery **263**
17.1 Introduction 263
17.2 Zigzag Relation 265
17.3 Communication-Induced Checkpointing 267
17.4 Optimistic Message Logging: Main Ideas 268
17.4.1 Model 269
17.4.2 Fault-Tolerant Vector Clock 270
17.4.3 Version End Table 272
17.5 An Asynchronous Recovery Protocol 272
17.5.1 Message Receive 274
17.5.2 On Restart after a Failure 274
17.5.3 On Receiving a Token 274
17.5.4 On Rollback 276
17.6 Problems 277
17.7 Bibliographic Remarks 278

18 Self-Stabilization **279**
18.1 Introduction 279
18.2 Mutual Exclusion with K-State Machines 280
18.3 Self-Stabilizing Spanning Tree Construction 285
18.4 Problems 286
18.5 Bibliographic Remarks 289

A Various Utility Classes **291**

Bibliography **291**

Index **304**