Contents

Preface **xvii**

Photo and Text Credits **xix**

Introduction **1**

Dahlia Malkhi, Idit Keidar

Chapter 1: Shared Memory and the Bakery Algorithm 3
Chapter 2: The Notions of Time and Global State in a Distributed System 5
Chapter 3: Byzantine Faults 7
Chapter 4: State Machine Replication with Benign Failures 9
Chapter 5: Formal Specification and Verification 10
Chapter 6: Biography 11
Closing Remarks 12

The Computer Science of Concurrency: The Early Years **13**

Leslie Lamport

1 Foreword 13
2 The Beginning: Mutual Exclusion 14
3 Producer-Consumer Synchronization 17
4 Distributed Algorithms 23
5 Afterwards 24
References 25

PART I TECHNICAL PERSPECTIVES ON LAMPORT'S WORK 27

Chapter 1 Shared Memory and the Bakery Algorithm 29

Hagit Attiya, Jennifer L. Welch

1.1 Introduction 29

1.2 Flavors of the Bakery Algorithm 30
1.3 A Plethora of Registers 36
1.4 A New Model for Describing Concurrency 41
1.5 Sequential Consistency 44

Chapter 2 The Notions of Time and Global State in a Distributed System 47

Karolos Antoniadis, Rachid Guerraoui

2.1 Introduction 47
2.2 The Notion of Logical Time 49
2.3 The Distributed State Machine Abstraction 53
2.4 The Notion of Distributed Global State 58
2.5 Conclusion 65

Chapter 3 Byzantine Faults 67

Christian Cachin

3.1 Introduction 67
3.2 Byzantine Agreement 68
3.3 Byzantine Clock Synchronization 76
3.4 Digital Signatures 77

Chapter 4 State Machine Replication with Benign Failures 83

Robbert van Renesse

4.1 Active versus Passive Replication 85
4.2 A Brief Review of State Machine Replication 85
4.3 Benign System Models 87
4.4 SMR Protocol Basics 88
4.5 Early Asynchronous Consensus Protocols 90
4.6 Paxos 96
4.7 Dynamic Reconfiguration 100

Chapter 5 Formal Specification and Verification 103

Stephan Merz

5.1 Introduction 103
5.2 The Temporal Logic of Actions 105
5.3 The Specification Language TLA$^+$ 115
5.4 PlusCal: An Algorithm Language 119

5.5 Tool Support 123
5.6 Impact 127

Chapter 6 Biography 131

Roy Levin

6.1 Early Years 131
6.2 Education and Early Employment 133
6.3 The COMPASS Years (1970–1977) 134
6.4 The SRI Years (1977–1985) 139
6.5 The DEC/Compaq Years (1985–2001) 146
6.6 The Microsoft Years (2001–) 155
6.7 Honors 163
6.8 Collegial Influences 165

PART II SELECTED PAPERS 171

A New Solution of Dijkstra's Concurrent Programming Problem 173

Leslie Lamport

Introduction 173
The Algorithm 174
Proof of Correctness 175
Further Remarks 177
Conclusion 177
References 178

Time, Clocks, and the Ordering of Events in a Distributed System 179

Leslie Lamport

Introduction 180
The Partial Ordering 180
Logical Clocks 182
Ordering the Events Totally 185
Anomalous Behavior 189
Physical Clocks 190
Conclusion 192
Appendix 193
References 196

How to Make a Multiprocessor Computer That Correctly Executes Multiprocess Programs 197

Leslie Lamport

References 201

The Byzantine Generals Problem 203

Leslie Lamport, Robert Shostak, Marshall Pease

1 Introduction 204
2 Impossibility Results 206
3 A Solution with Oral Messages 210
4 A Solution with Signed Messages 213
5 Missing Communication Paths 217
6 Reliable Systems 221
7 Conclusion 225
References 226

The Mutual Exclusion Problem: Part I—A Theory of Interprocess Communication 227

Leslie Lamport

Abstract 227
1 Introduction 228
2 The Model 228
3 Interprocess Communication 234
4 Processes 239
5 Multiple-Reader Variables 242
6 Discussion of the Assumptions 243
7 Conclusion 244
Acknowledgments 244
References 244

The Mutual Exclusion Problem: Part II—Statement and Solutions 247

Leslie Lamport

Abstract 247
1 Introduction 248
2 The Problem 249

3 The Solutions 258
4 Conclusion 273
References 275

The Part-Time Parliament 277
Leslie Lamport
1 The Problem 278
2 The Single-Decree Synod 281
3 The Multidecree Parliament 294
4 Relevance to Computer Science 304
Acknowledgments 316
References 316

References 319

Index 335

Biographies 343