
Contents

Part I Basic Concepts 1

Hardware and Applications; Synchronization; Program Properties

Chapter 1 Sequential Programming 7

1.1 Language Notation 7

Declarations; Statements; Procedures

1.2 Logic, Propositions, and Predicates 14

Formal Logical Systems; Propositions; Predicates

1.3 A Programming Logic 23

Axioms; Inference Rules

1.4 Proofs in Programming Logic 31

Proof Outlines; Equivalence and Simulation

1.5 Program Derivation 36

Weakest Preconditions; Weakest Preconditions of Statements;
Linear Search Revisited; Sorting

Historical Notes and References 47

Exercises 49

Chapter 2 Concurrency and Synchronization 57

2.1 Specifying Concurrent Execution 57

2.2 Atomic Actions and Synchronization 59

Fine-Grained Atomicity; Specifying Synchronization

2.3 Semantics of Concurrent Execution 64

2.4 Techniques for Avoiding Interference 67
Disjoint Variables; Weakened Assertions; Global Invariants;
Synchronization

2.5 Auxiliary Variables 78

2.6 Safety and Liveness Properties 81
Proving Safety Properties; Scheduling Policies and Fairness

Historical Notes and References 86

Exercises 89

Part II Shared Variables 95

Chapter 3 Fine-Grained Synchronization 97

3.1 The Critical Section Problem 98
A Coarse-Grained Solution; Spin Locks: A Fine-Grained Solution;
Implementing Await Statements

3.2 Critical Sections: Tie-Breaker Algorithm 107
A Coarse-Grained Solution; A Fine-Grained Solution;
An N -Process Solution

3.3 Critical Sections: Ticket Algorithm 112
Coarse-Grained Solution; Fine-Grained Solutions

3.4 Critical Sections: Bakery Algorithm 115
Coarse-Grained Solution; A Fine-Grained Solution

3.5 Barrier Synchronization 120
Shared Counter; Flags and Coordinators; Symmetric Barriers

3.6 Data Parallel Algorithms 127
Parallel Prefix Computations; Operations on Linked Lists;
Grid Computations; Synchronous Multiprocessors

3.7 On-The-Fly Garbage Collection 134
Problem Specification; Solution Outline;
A Coarse-Grained Solution; A Fine-Grained Solution

3.8 Implementing Processes 146
A Single-Processor Kernel; A Multiprocessor Kernel

Historical Notes and References 155

Exercises 159

Chapter 4 Semaphores 171**4.1 Notation and Semantics 172****4.2 Basic Uses and Programming Techniques 175**

Critical Sections: Changing Variables; Barriers: Signaling Events;
Producers and Consumers: Split Binary Semaphores;
Bounded Buffers: Resource Counting

4.3 Selective Mutual Exclusion 189

Dining Philosophers; Readers and Writers

4.4 General Condition Synchronization 197

Readers and Writers Revisited; The Technique of Passing the Baton;
Readers and Writers Solution; Alternative Scheduling Policies

4.5 Resource Allocation 204

Problem Definition and General Solution Pattern;
Shortest-Job-Next Allocation

4.6 Implementation 210**Historical Notes and References 212****Exercises 214****Chapter 5 Conditional Critical Regions 225****5.1 Notation and Semantics 226**

Examples; Inference Rules; Safety and Liveness Properties

5.2 Dining Philosophers Revisited 232**5.3 Readers/Writers Revisited 235**

Readers' Preference Solution; Writers' Preference Solution

5.4 Interprocess Communication 238

Bounded Buffer With Exclusive Access;
Bounded Buffer With Concurrent Access

5.5 Scheduling and Resource Allocation 242**5.6 Implementations 245**

Using Busy Waiting; Using Semaphores with Passing the Baton;
Using Semaphores with Rem's Algorithm; Using a Kernel

Historical Notes and References 254**Exercises 255**

Chapter 6 Monitors 263**6.1 Programming Notation 264**

Synchronization in Monitors;
Additional Operations on Condition Variables

6.2 Formal Semantics and Program Proofs 271

Axioms and Proof Obligations; A Procedure Call Inference Rule;
Safety and Liveness Properties;
An Example: Readers and Writers

6.3 Synchronization Techniques 283

Interval Timer: Covering Conditions and Priority Wait;
A Fair Semaphore: Passing the Condition;
The Sleeping Barber Problem: Rendezvous

6.4 Disk Scheduling: Program Structures 295

Scheduler as a Separate Monitor;
Scheduler as an Intermediary

6.5 Alternative Approaches to Synchronization 305

Alternative Signaling Disciplines;
Equivalence of the Signaling Disciplines;
Differences Between the Signaling Disciplines;
Alternatives to Mutual Exclusion;
Path Expressions

6.6 Implementations 319

Using Semaphores; Using a Kernel

Historical Notes and References 325**Exercises 329****Part III Message Passing 339****Chapter 7 Asynchronous Message Passing 343****7.1 Programming Notation 344****7.2 Formal Semantics 346**

Axioms and Satisfaction Proofs;
Auxiliary Variables and Non-Interference; An Example;
Safety and Liveness Properties

7.3 Filters: A Sorting Network 355

7.4 Clients and Servers 359

Active Monitors; A Self-Scheduling Disk Driver;
File Servers: Conversational Continuity

7.5 Heartbeat Algorithms 370

Network Topology: Shared-Variable Solution;
Network Topology: Distributed Solution

7.6 Probe/Echo Algorithms 376

Broadcast in a Network; Network Topology Revisited

7.7 Broadcast Algorithms 383

Logical Clocks and Event Ordering; Distributed Semaphores

7.8 Token-Passing Algorithms 388

Distributed Mutual Exclusion; Termination Detection in a Ring;
Termination Detection in a Graph

7.9 Replicated Servers 396

Replicated Files; Replicated Workers: Adaptive Quadrature

7.10 Implementations 402

Shared-Memory Kernel; Distributed Kernel

Historical Notes and References 410**Exercises 415****Chapter 8 Synchronous Message Passing 423****8.1 Programming Notation 424**

Communication Statements; Guarded Communication

8.2 Formal Semantics 429

Axioms, Inference Rules, and Satisfaction Proofs;
Auxiliary Variables and Non-Interference; An Example;
Safety and Liveness Properties

8.3 Networks of Filters 439

Prime Number Generation: The Sieve of Eratosthenes;
Matrix/Vector Multiplication

8.4 Interacting Parallel Processes 443

Parallel Sorting: A Heartbeat Algorithm;
Parallel Prefix Computations;
Matrix Multiplication: Broadcast Algorithm;
Matrix Multiplication: Heartbeat Algorithm

8.5 Clients and Servers 453

Resource Allocation; File Servers and Conversational Continuity;
Centralized Dining Philosophers; Decentralized Dining Philosophers

8.6 Implementations 460

Centralized Clearing House; Decentralized Implementations

Historical Notes and References 472**Exercises 474****Chapter 9 RPC and Rendezvous 483****9.1 Remote Procedure Call 484**

Synchronization in Modules; A Time Server;
Caches in a Distributed File System;
A Sorting Network of Merge Filters;
Formal Semantics

9.2 Rendezvous 494

Client/Server Examples; A Sorting Network of Merge Filters;
Formal Semantics; An Example Proof;
Safety and Liveness Properties

9.3 A Multiple Primitives Notation 509

Invoking and Servicing Operations; Examples; Formal Semantics

9.4 Clients and Servers 515

Readers/Writers Revisited: Encapsulated Access; Replicated Files;
Network Topology Using Probe/Echo Algorithms

9.5 Parallel Algorithms 522

Region Labeling: A Heartbeat Algorithm;
The Traveling Salesman Problem: Replicated Workers

9.6 Implementations 529

RPC in a Kernel; Rendezvous Using Asynchronous Message Passing;
Multiple Primitives in a Kernel

Historical Notes and References 540**Exercises 543**

Part IV Practice 551**Chapter 10 Language Overviews 553****10.1 Turing Plus: Monitors 555**

Program Structure; Process Interaction;
Example: File Difference Checker

10.2 Occam: Synchronous Message Passing 561

Program Structure; Sequential and Parallel Constructors;
Communication and Synchronization;
Example: Prime Number Sieve

10.3 Ada: Rendezvous 567

Program Components; Communication and Synchronization;
Example: The Dining Philosophers

10.4 SR: Multiple Primitives 575

Program Components; Communication and Synchronization;
Example: The Traveling Salesman Problem

10.5 Linda: Distributed Data Structures 581

Tuple Space and Process Interaction;
Example: Prime Numbers with Replicated Workers

10.6 Comparison and Performance 588

Performance Experiments; Interpretation of Results

Historical Notes and References 593**Exercises 598****Glossary 603****Bibliography 607****Index 627**