

OXFORD
HIGHER EDUCATION

Distributed Computing

SECOND EDITION



SUNITA MAHAJAN
SEEMA SHAH

Distributed Computing

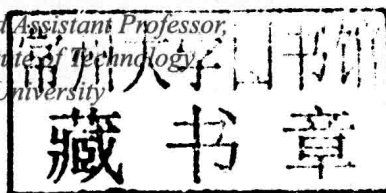
SECOND EDITION

Sunita Mahajan

*Principal, Institute of Computer Science
MET League of Colleges
Mumbai*

Seema Shah

*Acting Principal and Assistant Professor,
Vidyalankar Institute of Technology
Mumbai University*



OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Oxford University Press is a department of the University of Oxford. It furthers the University's objective of excellence in research, scholarship, and education by publishing worldwide. Oxford is a registered trade mark of Oxford University Press in the UK and in certain other countries.

Published in India by
Oxford University Press
YMCA Library Building, 1 Jai Singh Road, New Delhi 110001, India

© Oxford University Press 2010, 2013

The moral rights of the author/s have been asserted.

First Edition published in 2010
Second Edition published in 2013
Second impression 2013

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior permission in writing of Oxford University Press, or as expressly permitted by law, by licence, or under terms agreed with the appropriate reprographics rights organization. Enquiries concerning reproduction outside the scope of the above should be sent to the Rights Department, Oxford University Press, at the address above.

You must not circulate this work in any other form
and you must impose this same condition on any acquirer.

ISBN-13: 978-0-19-809348-0
ISBN-10: 0-19-809348-9

Typeset in Times New Roman
by Pee-Gee Graphics, New Delhi
Printed in India by Yash Printographics, Noida 201301

Third-party website addresses mentioned in this book are provided
by Oxford University Press in good faith and for information only.
Oxford University Press disclaims any responsibility for the material contained therein.

*Dedicated
to
our students,
colleagues, and family members*

Preface to the Second Edition

Initiated in 1982, the concept of distributed computing became popular with the advent of faster computers and high-speed networks, connecting disjoint but networked computers. Later using distributing applications over many computers, World Wide Web led to the sharing of databases, multimedia applications, and videoconferencing. These computers are either homogeneous or heterogeneous, connected together with high-speed communication network. The resources of the combined system are shared among different processes. The sharing and synchronization between different resources, such as memory pages, files, disk space, and databases, is done with the help of messages. Therefore, message communication, its optimization, security, and information consistency among various resources form an important portion of the distributed systems.

Computer science and computer engineering departments in a large number of universities introduce operating systems as an elementary course and move on to teaching distributed computing or distributed operating systems in subsequent semesters. Considerable amount of research has been done in distributed operating systems, database systems, and multiprocessor operating systems. Applications of distributed systems may be seen all around us, for example, in automated ticket bookings, purchasing goods online, handling large databases, etc. Core banking has been a major part of our lives where distributed computing is the underlying concept. Thus, it has become essential for students to study the concepts of distributed computing in detail due to its various applications in different industries.

About the Book

This text has been designed keeping in mind the requirements of students of computer science engineering, information technology, and computer applications. The book contains an exhaustive set of topics, common to all these departments. The presentation of the book is kept simple and stimulating with current case studies and detailed coverage of the topics. The highlight of the text is its simple, lucid language with numerous examples, illustrations, and cases.

The book presents an in-depth discussion on distributed computing fundamentals, interprocess communication, synchronization, distributed system management, distributed shared memory, distributed file system, and naming as well as grid computing, Service-Oriented Architecture (SOA), and cloud computing.

The key features of the book are as follows:

- Explains the theory diagrammatically or in tabulated form to help visualize the concepts
- Includes numerous objective and review questions to help understand the concepts taught in the chapters
- Covers important topics such as cloud computing and SOA

- Includes case studies within the chapters to facilitate the understanding of concepts discussed
- Includes appendices on Chorus, CORBA, and Mach operating systems at the end of the book

New to the Second Edition

This second edition includes a more detailed discussion on some of the concepts, additional application-oriented topics, and a theoretical formulation for distributed system concepts. The following are the major additions to this book:

- A new chapter on Formal Models that makes the formulation and implementation of various algorithms more systematic. It explains how to define a formal specification for one system simulating another system, thus leading to the best algorithm.
- Two new case studies on CORBA (a distributed object model using RMI concepts) including the Java implementation of the CORBA object model and Mach operating system (a distributed operating system).
- A new section on API for Internet protocol at the end of Chapter 3 that includes examples of how this can be implemented in Java, since Java has become the most prevalent computing language. Any interested user should be able to develop his/her own API with the help of these examples.

Contents and Coverage

The book is organized into 14 chapters.

Chapter 1 discusses the basic concepts of distributed systems, distributed computing models, software concepts, issues in designing distributed systems, client-server model, and current case studies of the World Wide Web 1.0, 2.0, and 3.0. *Chapter 2* provides information on the basics of computer networks, along with protocols and models for network systems, ATM systems, and distributed systems. *Chapter 3* introduces the fundamental concepts related to interprocess communication including message-passing mechanism, concepts of group communication, and case studies on IPC in Mach and CBCAST Protocol in ISIS. It also gives the application programming interface for Internet Protocols.

Chapter 4 presents the basic concepts of middleware, Remote Procedural Call (RPC), and Remote Method Invocation (RMI) along with case studies on Sun RPC and Java RMI. *Chapter 5* deals with clock synchronization, physical and logical clocks, global state, mutual exclusion algorithms, election algorithms, and deadlocks in distributed systems. *Chapter 6* defines a formal model for distributed system, communication systems, processes, and admissibility conditions and also gives the basics of simulation. *Chapter 7* focuses on the concepts related to resource management, process management, threads, and fault tolerance.

Chapter 8 discusses the fundamental concepts of Distributed Shared Memory (DSM), various hardware DSM systems, issues in designing and implementing DSM systems, heterogeneous and other DSM systems followed by a brief description of traditional DSM, such as Munin, TreadMarks, and Linda, along with prevalent DSM systems such as Teamster and JUMP. *Chapter 9*

includes the concepts of a Distributed File System (DFS), file models, issues in file system design, naming transparency and semantics of file sharing, techniques of DFS implementation, file caching and replication, and case studies on Sun Network File System and Google File System.

Chapter 10 describes all concepts related to naming, features and types of naming schemes, object locating mechanisms, issues in designing human-oriented names, name caches and naming related access control mechanisms, which define object security. The Domain Name Service (DNS) is also explained as a case study. *Chapter 11* introduces the basic concepts of security and cryptography, techniques for maintaining secure channels, and security in mobile codes, firewalls, and key management along with case studies on Kerberos and Electronic Payment System.

Chapter 12 describes the basic concepts of real-time distributed system, design issues, real-time communication, and real-time scheduling techniques along with a case study on real-time communication in MARS. *Chapter 13* presents the concepts of distributed database management systems and their architectures, and focuses on handling distributed query processing, replication, recovery in distributed environment, deadlock detection, and modified procedures like two- and three-phase commit. This chapter also introduces mobile databases and discusses Oracle's distributed functionality as a case study.

Chapter 14 explains the emerging trends in computing, namely grid computing, service-oriented computing, and cloud computing. It discusses the use of simulators, grid-building toolkits, and the grids available on the Internet. This chapter further describes Service-Oriented Architecture (SOA), its relation with web services, service-oriented grid, SOA design, and its advantages. Cloud computing with its architecture and landscape is also introduced in this chapter. *Appendices A, B, and C* provide discussions on Chorus, CORBA, and Mach operating systems, respectively.

Acknowledgements

We would like to thank our colleagues for using it as a textbook in their distributed computing lectures. Their valuable suggestions were extremely useful while developing the new material.

We express our sincere gratitude and thanks to the entire editorial team of Oxford University Press for bringing out this second edition.

We thank all who have helped us directly or indirectly during the project. We have made every effort to make the book student-friendly and error-free. Despite this, some errors may have crept in inadvertently. We will be grateful if they are brought to notice via email at sunitam_ics@met.edu or sunitamm@gmail.com or 4.seema@gmail.com.

Sunita Mahajan

Seema Shah

Preface to the First Edition

Distributed computing is the study of the way distributed systems compute distributed applications. A distributed system consists of multiple high performance computers connected by equally high-speed communication links working as a single consolidated system. In such a system, multiple resources work together to deliver the required processing speed and the distributed operating system takes care of maintaining the overall system.

Distributed computing and distributed systems are terms often used interchangeably. In distributed systems, the focus is on building a system, while distributed computing deals with its applicability. Gradually, the focus is shifting towards the applications part of distributed systems. Considerable amount of research has been undertaken in distributed operating systems, database operating systems, and multiprocessor operating systems. The field is rapidly changing and is now considered as one that belongs to the mainstream. This book focuses on the application aspects while providing a strong background of the systems part. Areas such as distributed databases and emerging areas of computing like grid and ubiquitous computing are also gaining importance. These and other emerging areas are discussed at length in this book.

This book provides cohesive information about distributed systems: its concepts and applications. The topics covered in this book would form the basis of an advanced course on operating systems and distributed computing.

About the Book

The contents of this book were finalized after studying the curriculum of many universities in India and abroad to determine the requirements in graduate and postgraduate level courses in operating systems and distributed systems. The chapters have been developed from the lecture notes used by the authors while teaching advanced operating systems, distributed operating systems, and distributed computing courses at various Indian universities over the past several years. The chapters of the book include many examples and illustrations that complement the text. All chapters contain review questions and multiple choice questions.

The presentation of topics in the book has been kept simple and stimulating with current case studies in most chapters. There is a separate case study at the end of the book that illustrates the practical implementation of the concepts of distributed systems covered in the book. The book also presents the emerging trends in distributed computing in a separate chapter. A detailed bibliography is provided at the end of the book for suggested further reading so that interested readers can explore topics beyond the scope of this book.

The key features of the book are as follows:

- Explains the theory diagrammatically and in tabulated form to help visualize the concepts
- Includes practice questions and MCQs to help review the concepts taught in the chapters
- Covers emerging topics such as grid computing, cloud computing, and service-oriented architecture
- Includes case studies within the chapters to facilitate the understanding of concepts discussed
- Includes a separate case study at the end of the book on Chorus operating system

Acknowledgements

An effort to develop a text of this kind is a group activity and it calls for generous help and support from a lot of people. We take this opportunity to express our gratitude to all without whom this text would have been a distant dream.

We express our sincere gratitude to the entire editorial team of Oxford University Press for publishing our manuscript and helping us overcome all hurdles and difficulties faced during the implementation of this project.

Finally, we thank all who have helped us directly or indirectly during the project and especially our family members who supported us during this project.

Sunita Mahajan

Seema Shah

Features of the Book

Learning Objectives

Learning objectives at the beginning of each chapter provide an outline of the topics to be discussed in the chapter.

Learning Objectives

This chapter will enable you to understand:

- Clock synchronization including physical clocks and clock synchronization algorithms
- Logical clocks which comprises event ordering, Lamport's timestamps, and vector timestamps implementation, strengths, and weaknesses
- Mutual exclusion algorithms: centralized algorithm, distributed algorithm, Token ring algorithm, and their comparison

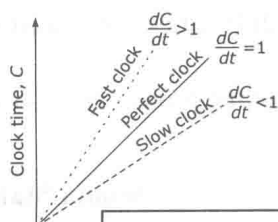


Figure 5-2 Clock

Table 12-1 Dynamic vs static scheduling

Dynamic scheduling	Static scheduling
Suitable for event-triggered design	Good choice for time-
Scheduling decisions are made on-the-fly	Needs to be carefully
Better utilization of resources	Over-estimation may
No time available for making complex	There is time to select
scheduling calculations during execution	

Tables and Figures

Numerous tables and figures present throughout the book help in visualization of the concepts discussed.

Notes

Notes present across all chapters provide important points at the end of the topics discussed for a quick review.



Real-time scheduling is based on deciding between various parameters, such as hard real-time versus soft real-time, pre-emptive versus non-pre-emptive, dynamic versus static scheduling, and centralized versus decentralized scheduling.

CASE STUDY

Real-time Communication in MARS

The MARS real-time system uses the Time-Triggered Protocol (TTP). A MAR CPUs which together present a single fault-tolerant fail-silent image. MARS no using two reliable and independent TDMA broadcast networks. The expected every 30 million years because all packets are sent to both the networks in par

Case Study

Case studies present within relevant chapters facilitate the understanding of the concepts discussed in the chapters.

Brief Contents

Preface to the First Edition v

Preface to the Second Edition viii

Features of the Book x

Detailed Contents xiii

Chapter 1 Basic Distributed System Concepts	1
Chapter 2 Network Communication	40
Chapter 3 Interprocess Communication	61
Chapter 4 Remote Communication	100
Chapter 5 Synchronization	141
Chapter 6 Simulation Basics for Formal Model Specifications	210
Chapter 7 Distributed System Management	225
Chapter 8 Distributed Shared Memory	273
Chapter 9 Distributed File System	316
Chapter 10 Naming	349
Chapter 11 Security in Distributed Systems	384
Chapter 12 Real-time Distributed Operating Systems	450
Chapter 13 Distributed Database Management System	464
Chapter 14 Emerging Trends in Distributed Computing	516
<i>Appendix A Case Study: Chorus</i>	<i>554</i>
<i>Appendix B Case Study: CORBA</i>	<i>566</i>
<i>Appendix C Case Study: Mach</i>	<i>586</i>
<i>Web References</i>	<i>609</i>
<i>Index</i>	<i>611</i>

Summary

Real-time systems respond to external stimulus within a deadline. Both the correctness and the instant of response are equally important in a real-time system. There are three types of stimuli: periodic, aperiodic, and sporadic, and the real-time system responds to each of them separately. A soft real-time system is allowed to

packet transmission in advance. upper bound can be easily specified. Multiple Access (TDMA) ensures transmitted in fixed size frames with n slots. This scheme ensures that a fraction of the bandwidth, delay

Summary

Chapter-end summary provides a quick recap of the detailed discussions from the chapter, thus offering a lead to recall them.

Objective and Review Questions

Objective questions (multiple choice questions, true/false, and fill in the blanks) and review questions help students test their understanding of the topics discussed.

EXERCISE

Objective Questions

Select the correct option(s).

1. A network is a _____ of channels.
 - (a) Collection
 - (b) Group
 - (c) None of the above
2. The basic methods of message passing are _____ and _____.
 - (a) Asynchronous, synchronous

- (c) Computational event
- (d) Delivery event
- (e) Integrity
- (f) Schedule of execution

Review Questions

1. Differentiate between asynchronous and synchronous message passing.

Appendix A

A.1 Introduction

Chorus is a microkernel-based distributed operating system that started as a research project in 1979 at INRIA (Institut National de Recherche en Informatique et Automatique) laboratory in

Case Study

distributed virtual machines were borrowed from capabilities for global was borrowed from A

Appendices A, B, and C

Three appendices at the end of the book are dedicated to Chorus, CORBA, and Mach operating systems.

Appendix B

B.1 Introduction

Object-oriented concepts are useful for developing non-distributed applications. The object interface hides the internal details of the external world. The design of a program that

Case Study

- Maybe semantics: not know what executed once

Appendix C

C.1 Introduction

Mach has evolved from the communication system and philosophy of the Accent operating system. It was developed at the Carnegie Mellon University (CMU). However, the virtual memory system and the management of tasks

Case Study

shared between processes can be easily ported to other architectures. A key goal of the system was the capability to support heterogeneous hardware and operating systems and allowin

Detailed Contents

Preface to the First Edition v

Preface to the Second Edition viii

Features of the Book x

Brief Contents xii

1. Basic Distributed System Concepts 1

- 1.1 Introduction 2
 - 1.1.1 What is a Distributed System? 2
 - 1.1.2 Architectures for Distributed Systems 4
- 1.2 Distributed Computing Models 6
 - 1.2.1 Workstation Model 6
 - 1.2.2 Workstation-Server Model 7
 - 1.2.3 Processor-pool Model 8
 - 1.2.4 Comparison of the Distributed Computing Models 9
 - 1.2.5 Advantages of Distributed Systems 9
 - 1.2.6 Disadvantages of Distributed Systems 12
- 1.3 Software Concepts 13
 - 1.3.1 Network Operating System 14
 - 1.3.2 Distributed Operating System 15
 - 1.3.3 Multiprocessor Time-sharing System 16
 - 1.3.4 Comparison of Operating Systems 18
- 1.4 Issues in Designing Distributed Systems 18
 - 1.4.1 Transparency 18
 - 1.4.2 Flexibility 21
 - 1.4.3 Reliability 23
 - 1.4.4 Performance 23
 - 1.4.5 Scalability 24
 - 1.4.6 Security 26
 - 1.4.7 Fault Tolerance 26

- 1.5 Client-Server Model 26
 - 1.5.1 Basic Concepts 27
 - 1.5.2 Client-Server Addressing 27
 - 1.5.3 Client-Server Implementation 29
 - 1.5.4 Client-Server Architecture 32

Case Studies:

- World Wide Web 1.0* 33
- World Wide Web 2.0* 35
- World Wide Web 3.0* 37

2. Network Communication 40

- 2.1 LAN and WAN Technologies 41
 - 2.1.1 Introduction to LAN and WAN 41
 - 2.1.2 Classification of Networks 41
- 2.2 Protocols for Network Systems 45
 - 2.2.1 The ISO/OSI Reference Model 46
 - 2.2.2 Internet Protocols 49
- 2.3 Asynchronous Transfer Mode 51
 - 2.3.1 Introduction to ATM 51
 - 2.3.2 ATM Protocol Reference Model 52
- 2.4 Protocols for Distributed Systems 54
 - 2.4.1 FLIP 55
 - 2.4.2 VMTP 57

3. Interprocess Communication 61

- 3.1 Message Passing 62

3.1.1	Introduction to Message Passing	62	4.5.1	Exception Handling and Security	119
3.1.2	Advantages and Features of Message-passing Systems	62	4.5.2	RPC in Heterogeneous Environment	119
3.1.3	IPC Message Format	65	4.5.3	Failure Handling	120
3.1.4	IPC Synchronization	66	4.5.4	RPC Optimization	121
3.1.5	Message Buffering Strategies	68	4.5.5	Complicated and Special RPCs	124
3.1.6	Multidatagram Messaging	70	<i>Case Study: Sun RPC</i>		127
3.1.7	Process Addressing Techniques	72	4.6	Remote Method Invocation Basics	128
3.1.8	Failure Handling Mechanism	74	4.6.1	Distributed Object Concepts	128
<i>Case Study: IPC in Mach</i>		76	4.7	RMI Implementation	130
3.2	Group Communication	78	4.7.1	Design Issues in RMI	130
3.2.1	Types of Group Communication	78	4.7.2	RMI Execution	132
3.2.2	Group Management	80	4.7.3	Types of Objects	134
3.2.3	Group Addressing and Message Delivery	80	4.7.4	Binding a Client to an Object	134
3.2.4	Reliability Mechanism	81	4.7.5	RMI Parameter Passing	135
3.2.5	Message Ordering	83	<i>Case Study: Java RMI</i>		136
<i>Case Study: CBCAST Protocol in ISIS</i>		85	5. Synchronization		141
3.3	API for Internet Protocol	87	5.1	Introduction	142
3.3.1	Synchronous and Asynchronous Communication	87	5.2	Clock Synchronization	142
3.3.2	Sockets, UDP, and TCP	88	5.2.1	Physical Clocks	142
3.3.3	Java API for UDP and TCP Protocols	93	5.2.2	Clock Synchronization Algorithms	147
			5.2.3	Use of Synchronized Clocks	153
4. Remote Communication		100	5.3	Logical Clocks	155
4.1	Introduction to Remote Communication	101	5.3.1	Event Ordering	155
4.1.1	Middleware	101	5.3.2	Implementation of Logical Clocks	156
4.2	Remote Procedural Call Basics	102	5.3.3	Lamport's Timestamps	158
4.2.1	Basic RPC Operation	102	5.3.4	Vector Timestamps	161
4.2.2	Stub Generation in RPC	105	5.4	Global State	163
4.3	RPC Implementation	105	5.5	Mutual Exclusion	166
4.3.1	RPC Messages	106	5.5.1	Centralized Algorithm	167
4.3.2	Parameter Passing Semantics	108	5.5.2	Distributed Algorithm	168
4.3.3	Server Management	111	5.5.3	Token Ring Algorithm	169
4.4	RPC Communication	112	5.5.4	Comparison of Various Algorithms	170
4.4.1	RPC Call Semantics	112	5.6	Election Algorithms	171
4.4.2	RPC Communication Protocols	114	5.6.1	Bully Election Algorithm	172
4.4.3	Client-Server Binding	116	5.6.2	Ring Election Algorithm	173
4.5	Other RPC Issues	118	5.6.3	Election in a Wireless Network	175
			5.7	Deadlocks in Distributed Systems	177
			5.7.1	Basic Concepts	177

5.7.2	Deadlock Modelling	180
5.7.3	Handling Deadlocks in Distributed Systems	185
5.7.4	Distributed Deadlock Prevention	191
5.7.5	Distributed Deadlock Detection	195
5.7.6	Distributed Deadlock Recovery	204
5.7.7	Issues in Recovery from Deadlock	205
<i>Case Study: Deadlocks in Message Communication</i>		206

6. Simulation Basics for Formal Model Specifications 210

6.1	Introduction	211
6.1.1	Basic Model of a Message-passing System	211
6.2	Basics of Formal Model Specifications	215
6.2.1	A Generic Problem Specification	215
6.3	Communication Systems	216
6.3.1	Asynchronous Point-to-point Message-passing System	216
6.3.2	Asynchronous Broadcast	217
6.4	Processes	217
6.5	Admissibility Conditions	220
6.6	Simulations	221

7. Distributed System Management 225

7.1	Introduction	226
7.2	Resource Management	226
7.2.1	Desirable Features of a Global Scheduling Algorithm	227
7.3	Task Assignment Approach	228
7.3.1	Graph Theoretic Deterministic Algorithm	229
7.3.2	Centralized Heuristic Algorithm	230
7.3.3	Hierarchical Algorithm	231
7.4	Load-balancing Approach	232
7.4.1	Taxonomy of Load-balancing Algorithms	232
7.4.2	Issues in Designing Load-balancing Algorithms	233

7.5	Load-sharing Approach	240
7.5.1	Issues in Designing Load-sharing Algorithms	240
7.6	Process Management in a Distributed Environment	242
7.6.1	Functions of Distributed Process Management	243
7.6.2	Desirable Features of a Process Migration Mechanism	243
7.7	Process Migration	244
7.7.1	Mechanism for Process Migration	245
7.7.2	Address Space Transport Mechanisms	247
7.7.3	Message Forwarding Mechanism	249
7.7.4	Process Migration in Heterogeneous Systems	251
7.7.5	Advantages of Process Migration	253
7.8	Threads	254
7.8.1	Process vs Threads	254
7.8.2	Thread Models	255
7.8.3	Design Issues in Threads	257
7.8.4	Implementing a Thread Package	260
7.8.5	Threads and Remote Execution	262
7.9	Fault Tolerance	264
7.9.1	Component Faults	264
7.9.2	System Failures	265
7.9.3	Use of Redundancy	266

8. Distributed Shared Memory 273

8.1	Introduction	274
8.2	Basic Concepts of DSM	274
8.2.1	DSM Architecture	274
8.2.2	Message Passing vs Shared Memory	276
8.2.3	Types of DSMs	277
8.2.4	Advantages of DSM	278
8.3	Hardware DSM	279
8.3.1	On-chip Memory DSM	279
8.3.2	Bus-based Multiprocessor	280
8.3.3	Ring-based Multiprocessor	281

8.3.4 Switched Multiprocessor	282	9.6 File Caching in DFS	333
8.4 Design Issues in DSM Systems	283	9.6.1 Cache Location	333
8.4.1 Granularity	283	9.6.2 Cache Consistency	336
8.4.2 Structure	284	9.7 Replication in DFS	338
8.4.3 Consistency Models	285	9.7.1 Unit of Replication	338
8.4.4 Coherence Protocols	293	9.7.2 Replica Creation	339
8.5 Issues in Implementing DSM Systems	296	9.7.3 Update Protocols	340
8.5.1 Thrashing	296	<i>Case Studies:</i>	
8.5.2 Responsibility for DSM Management	297	<i>Sun Network File System</i>	341
8.5.3 Replication vs Migration Strategies	300	<i>Google File System</i>	345
8.5.4 Replacement Strategy	301		
8.6 Heterogeneous and Other DSM Systems	303	10. Naming	349
8.6.1 Data Compatibility and Conversion	303	10.1 Introduction	350
8.6.2 Block Size Selection	304	10.2 Desirable Features of a Good Naming System	350
<i>Case Studies:</i>		10.3 Basic Concepts	352
<i>Munin</i>	306	10.3.1 Name	352
<i>Linda</i>	307	10.3.2 Namespace and Contexts	354
<i>Teamster</i>	309	10.3.3 Name Server	358
<i>JUMP</i>	311	10.3.4 Name Agent	359
		10.3.5 Name Resolution	359
9. Distributed File System	316	10.4 System-oriented Names	360
9.1 Introduction to DFS	317	10.4.1 Features of System-oriented Names	360
9.1.1 Functions of DFS	317	10.4.2 Types of System-oriented Names	361
9.1.2 Components of DFS	317	10.4.3 Approaches to Generate System-oriented Names	361
9.1.3 Desirable Features of a Good DFS	318	10.5 Object-locating Mechanisms	363
9.2 File Models	320	10.5.1 Broadcast	364
9.2.1 Structured and Unstructured Files	320	10.5.2 Expanding Ring Broadcast	364
9.2.2 Mutable and Immutable Files	320	10.5.3 Encoding Object Location in UID	365
9.3 DFS Design	321	10.5.4 Encoding Creator Node ID in UID	366
9.3.1 File Service Interface	321	10.5.5 Using Forward Location Pointers	366
9.3.2 Directory Service Interface	323	10.5.6 Using Hint Cache and Broadcasting	367
9.3.3 Naming Transparency	325	10.6 Issues in Designing Human-oriented Names	368
9.4 Semantics of File Sharing	327	10.6.1 Scheme for Global Object Naming	369
9.4.1 UNIX File Semantics	328	10.6.2 Schemes for Partitioning a Namespace into Contexts	370
9.4.2 Session Semantics	328	10.6.3 Schemes for Implementing Context Bindings	371
9.4.3 Immutable Shared-file Semantics	329		
9.4.4 Transaction-like Semantics	329		
9.5 DFS Implementation	330		
9.5.1 DFS System Structure	330		