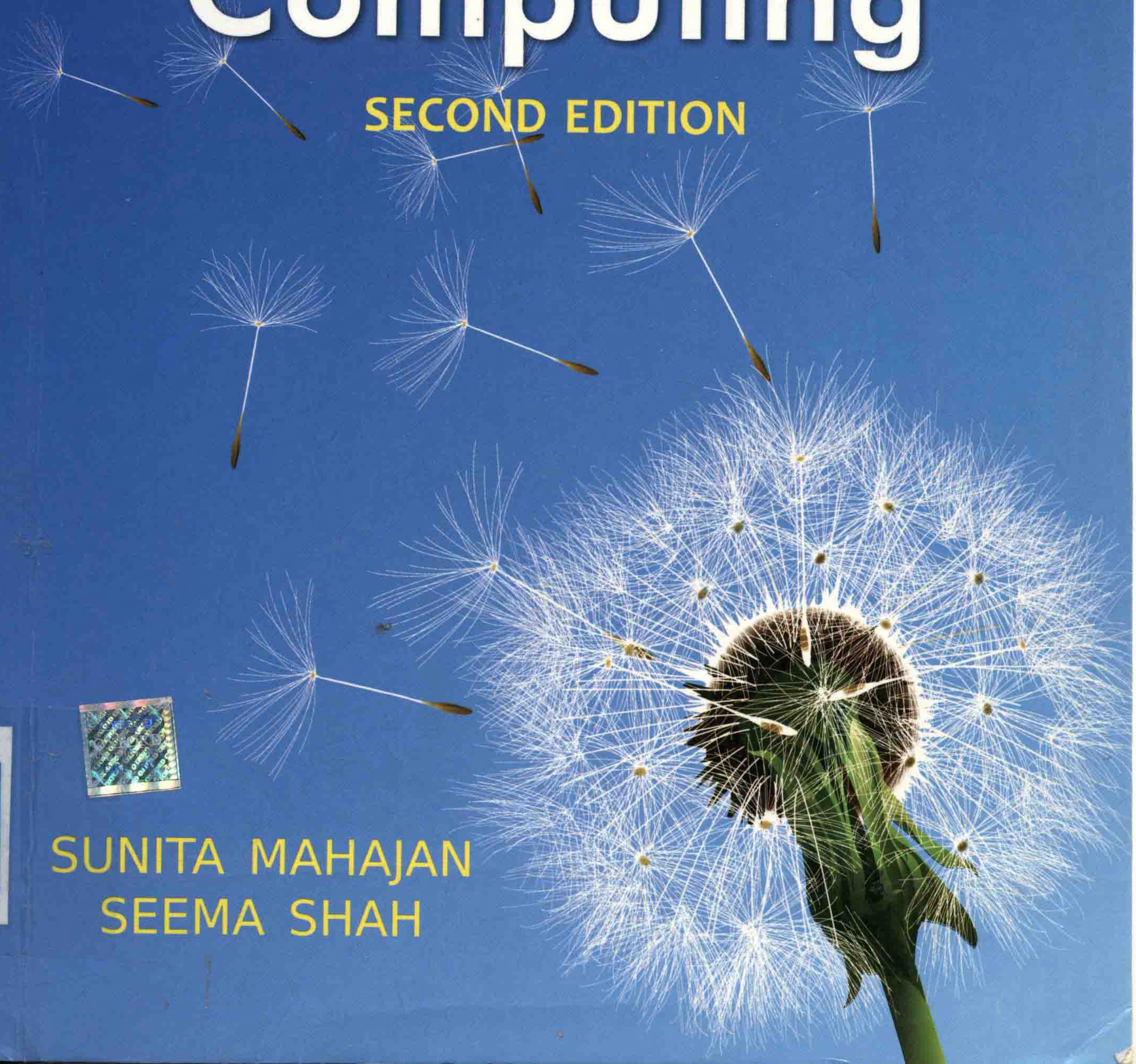
OXFORD
HIGHER EDUCATION

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

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

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

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.

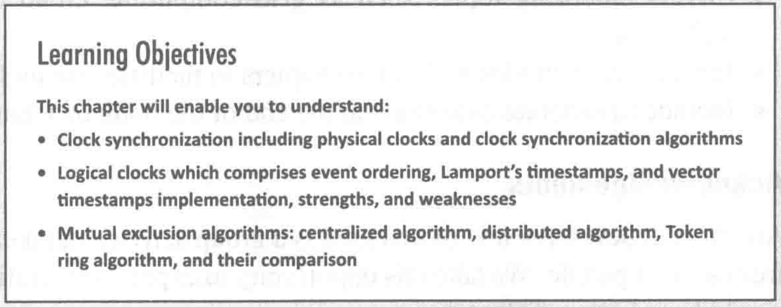
Sunita Mahajan

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**

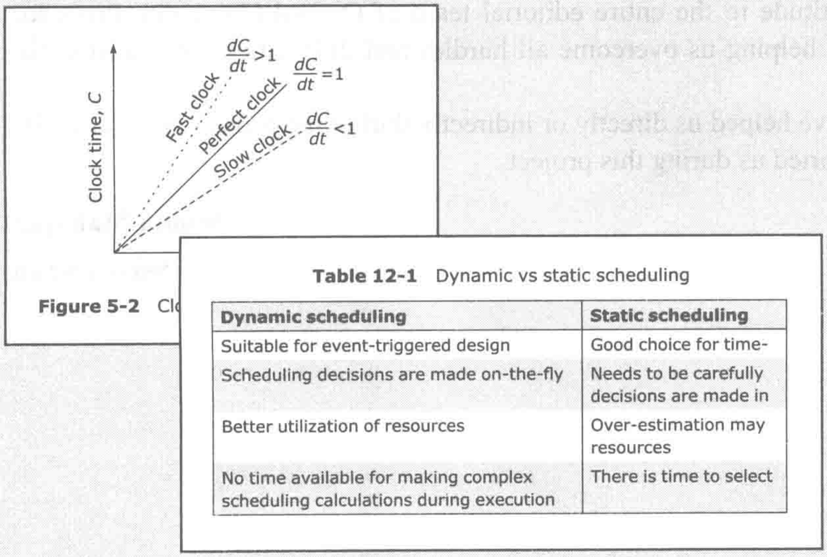


Figure 5-2 Cl

Table 12-1 Dynamic vs static scheduling

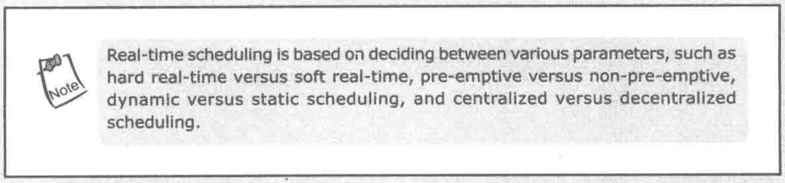
Dynamic scheduling	Static scheduling
Suitable for event-triggered design	Good choice for time-
Scheduling decisions are made on-the-fly	Needs to be carefully decisions are made in
Better utilization of resources	Over-estimation may resources
No time available for making complex scheduling calculations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

Tables and Figures

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

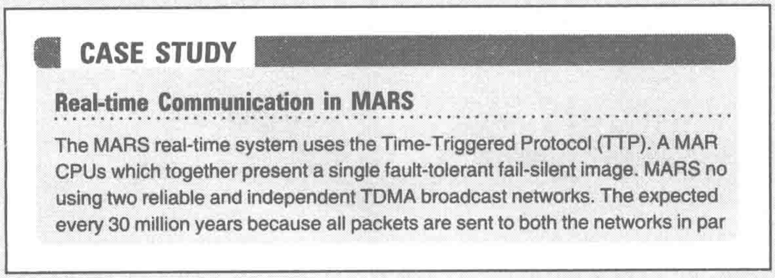
Notes

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



Note

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

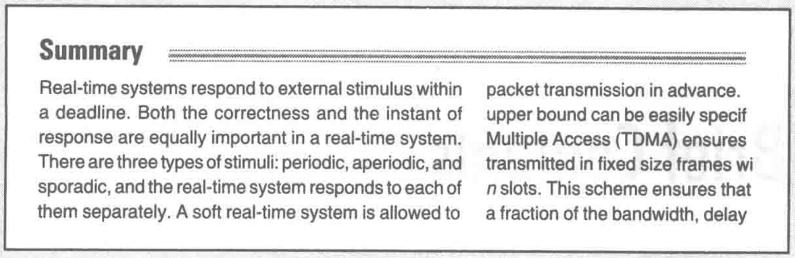
Real-time Communication in MARS

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



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

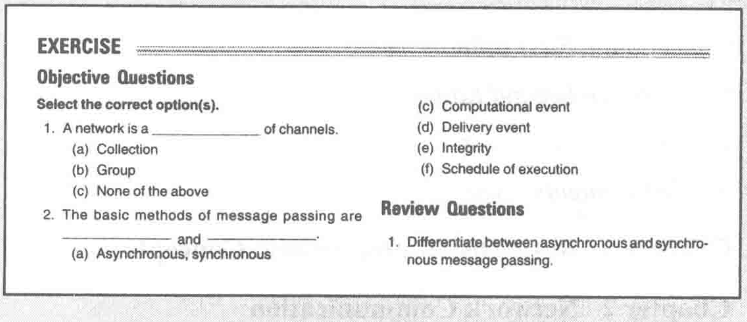
packet transmission in advance. upper bound can be easily specif Multiple Access (TDMA) ensures transmitted in fixed size frames wi *n* slots. This scheme ensures that a fraction of the bandwidth, delay

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

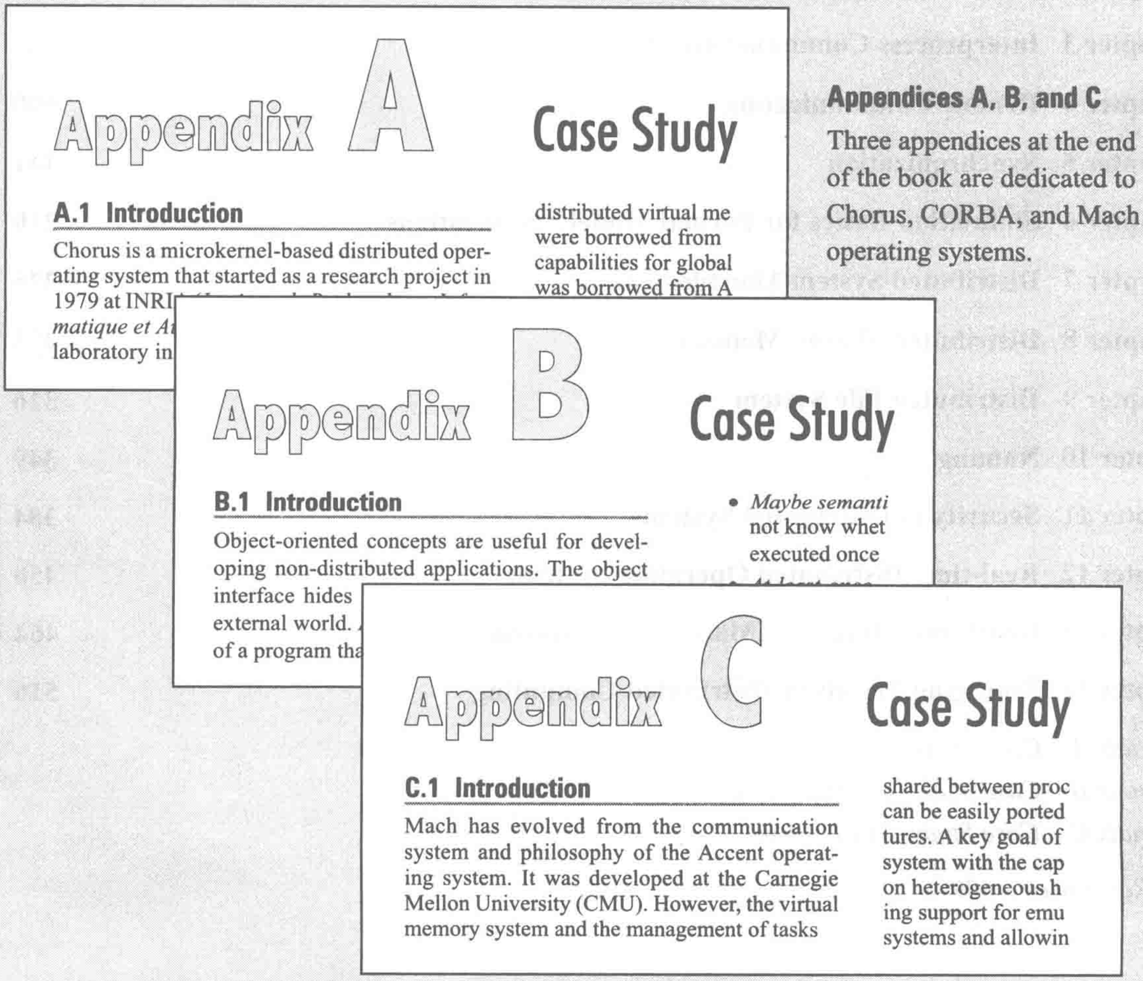
Objective Questions

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 Case Study

A.1 Introduction

Chorus is a microkernel-based distributed operating system that started as a research project in 1979 at INRI

matique et A

laboratory in

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

Appendix B Case Study

B.1 Introduction

Object-oriented concepts are useful for developing non-distributed applications. The object interface hides

external world.

of a program tha

- *Maybe semanti* not know whet executed once

Appendix C Case Study

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

shared between proc can be easily ported tures. A key goal of system with the cap on heterogeneous h ing support for emu systems and allowin

Appendices A, B, and C

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

Detailed Contents

Case Studies:

Case Studies: