大学计算机教育丛书（影印版）

Distributed Operating Systems

Andrew S. Tanenbaum

分布式操作系统

清华大学出版社 • PRENTICE HALL

Distributed Operating Systems

分布式操作系统

Andrew S. Tanenbaum

清华大学出版社
Prentice-Hall International, Inc.

(京)新登字 158 号

北京市版权局著作权合同登记号:01-97-0167

图书在版编目(CIP)数据

分布式操作系统:英文/(美)特南鲍姆(Tanenbaum, A. S.)著. -北京:清华大学出版社,1996.12
(大学计算机教育丛书:影印版)
ISBN 7-302-02411-1

Ⅰ.分… Ⅱ.特… Ⅲ.分布式操作系统-高等学校-教材-英文
Ⅳ.TP316

中国版本图书馆 CIP 数据核字(96)第 25162 号
出版者:清华大学出版社(北京清华大学学研楼,邮编 100084)
http:// www.tup.tsinghua.edu.cn
印刷者:清华大学印刷厂
发行者:新华书店总店北京发行所
开 本:850×1168 1/32 印张:19.75
版 次:1997 年 2 月第 1 版 2000 年 2 月第 6 次印刷
书 号:ISBN 7-302-02411-1/TP·1213
印 数:16001~19000
定 价:30.00 元

出版者的话

今天，我们的大学生、研究生和教学、科研工作者，面临的是一个国际化的信息时代。他们将需要随时查阅大量的外文资料；会有更多的机会参加国际性学术交流活动；接待外国学者；走上国际会议的讲坛。作为科技工作者，他们不仅应有与国外同行进行口头和书面交流的能力，更为重要的是，他们必须具备极强的查阅外文资料获取信息的能力。有鉴于此，在国家教委所颁布的“大学英语教学大纲”中有一条规定：专业阅读应作为必修课程开设。同时，在大纲中还规定了这门课程的学时和教学要求。有些高校除开设“专业阅读”课之外，还在某些专业课拟进行英语授课。但教、学双方都苦于没有一定数量的合适的英文原版教材作为教学参考书。为满足这方面的需要，我们陆续精选了一批国外计算机科学方面最新版本的著名教材，进行影印出版。我社获得国外著名出版公司和原著作者的授权将国际先进水平的教材引入我国高等学校，为师生们提供了教学用书，相信会对高校教材改革产生积极的影响。

我们欢迎高校师生将使用影印版教材的效果、意见反馈给我们，更欢迎国内专家、教授积极向我社推荐国外优秀计算机教育教材，以利我们将《大学计算机教育丛书(影印版)》做得更好，更适合高校师生的需要。

清华大学出版社

《大学计算机教育丛书(影印版)》项目组

1999.6

Preface

With the publication of *Distributed Operating Systems* I have now completed my trilogy on operating systems. The three volumes of this trilogy are:

- *Operating Systems: Design and Implementation*
- *Distributed Operating Systems*
- *Modern Operating Systems*

The three volumes are not completely independent, however. For schools having a two-course sequence in operating systems (or an undergraduate course plus a graduate course), one possible choice is to use *Operating Systems: Design and Implementation* in the first course and *Distributed Operating Systems* in the second one.

The former book treats the standard principles of single-processor systems, including processes, synchronization, I/O, deadlocks, memory management, file systems, security, and so on. It also illustrates these principles in great detail through the use of MINIX, a UNIX-clone whose source listing is given in an appendix. MINIX is available on diskette from Prentice Hall for the IBM PC (8088 and up), Atari, Amiga, Macintosh, and SPARC processors.

The latter book (this one), covers distributed operating systems in detail, including communication, synchronization, processes, file systems, and memory management, but this time in the context of distributed systems. Four examples of distributed systems are given in great detail: Amoeba, Mach, Chorus, and DCE. Amoeba is available for free to universities for educational use. It runs

on the Intel 386/486, SPARC, and Sun 3 processors. For information on how to obtain Amoeba please FTP the file *amoeba/Intro.ps.Z* from *ftp.cs.vu.nl* or contact the author by electronic mail at *ast@cs.vu.nl*. Potential users should be forewarned that Amoeba is considerably more complex than MINIX: the documentation alone (available by FTP), runs to well over 1000 pages and the system requires at least five large machines and an Ethernet to run well.

By studying these two books in sequence and using both MINIX and Amoeba, students will obtain a thorough knowledge of the principles and practice of both single-processor and distributed operating systems. Now that the trilogy is completed, I plan to revise MINIX and the book describing it.

For universities or computer professionals with less time available, *Modern Operating Systems* can be thought of as a condensed version of the other two books. It provides an introduction to the principles of both single-processor and distributed operating systems, but without the detailed example of MINIX. It also omits many of the advanced topics present in this book, including an introduction to ATM, fault-tolerant distributed systems, real time distributed systems, distributed shared memory, Chorus, DCE, and other topics. In all, about 230 pages of material on distributed systems present in this book have been omitted from *Modern Operating Systems*.

Many people have helped me with this book. I would especially like to thank the following people for reading portions of the manuscript and giving me many useful suggestions for improvement: Irina Athanasiu, Henri Bal, Saniya Ben Hassen, David Black, John Carter, Randall Dean, Wiebren de Jonge, John Dugas, Dick Grune, Anoop Gupta, Frans Kaashoek, Marcus Koebler, Hermann Kopetz, Ed Lazowska, Dan Lenoski, Kai Li, Marc Maathuis, David Mosberger, Douglas Orr, Craig Partridge, Carlton Pu, Marc Rozier, Rich Salz, Mike Schroeder, Karsten Schwan, Greg Sharp, Dennis Shasha, Sol Shatz, Jennifer Steiner, Chuck Thacker, John Turek, Walt Tuvell, Leendert van Doorn, Robbert van Renesse, Kees Verstoep, Ellen Zegura, Willy Zwaenpoel, and the anonymous reviewers. My editor, Bill Zobrist, put up with my attempts to get everything perfect with nary a whimper.

Despite all this help, no doubt some errors remain. That seems to be inevitable, no matter how many people read the manuscript. People who wish to report errors should contact me by electronic mail.

Finally, I would like to thank Suzanne again. After eight books, she knows the implications of another one, but her patience and love are boundless. I also want to thank Barbara and Marvin for using their computers and leaving mine alone (except for the printer). Teaching them how to use PC word processing programs has made me appreciate *troff* more than ever. Finally, I would like to thank Little Bram for being quiet while I was writing.

Andrew S. Tanenbaum

Contents

1

Introduction to Distributed Systems

Computer systems are undergoing a revolution. From 1945, when the modern computer era began, until about 1985, computers were large and expensive. Even minicomputers normally cost tens of thousands of dollars each. As a result, most organizations had only a handful of computers, and for lack of a way to connect them, these operated independently from one another.

Starting in the mid-1980s, however, two advances in technology began to change that situation. The first was the development of powerful microprocessors. Initially, these were 8-bit machines, but soon 16-, 32-, and even 64-bit CPUs became common. Many of these had the computing power of a decent-sized mainframe (i.e., large) computer, but for a fraction of the price.

The amount of improvement that has occurred in computer technology in the past half century is truly staggering and totally unprecedented in other industries. From a machine that cost 10 million dollars and executed 1 instruction per second, we have come to machines that cost 1000 dollars and execute 10 million instructions per second, a price/performance gain of 10^{11}. If cars had improved at this rate in the same time period, a Rolls Royce would now cost 10 dollars and get a billion miles per gallon. (Unfortunately, it would probably also have a 200-page manual telling how to open the door.)

The second development was the invention of high-speed computer networks. The **local area networks** or **LANs** allow dozens, or even hundreds, of machines within a building to be connected in such a way that small amounts of

information can be transferred between machines in a millisecond or so. Larger amounts of data can be moved between machines at rates of 10 to 100 million bits/sec and sometimes more. The **wide area networks** or **WANs** allow millions of machines all over the earth to be connected at speeds varying from 64 Kbps (kilobits per second) to gigabits per second for some advanced experimental networks.

The result of these technologies is that it is now not only feasible, but easy, to put together computing systems composed of large numbers of CPUs connected by a high-speed network. They are usually called **distributed systems,** in contrast to the previous **centralized systems** (or **single-processor systems**) consisting of a single CPU, its memory, peripherals, and some terminals.

There is only one fly in the ointment: software. Distributed systems need radically different software than centralized systems do. In particular, the necessary operating systems are only beginning to emerge. The first few steps have been taken, but there is still a long way to go. Nevertheless, enough is already known about these distributed operating systems that we can present the basic ideas. The rest of this book is devoted to studying concepts, implementation, and examples of distributed operating systems.

1.1. WHAT IS A DISTRIBUTED SYSTEM?

Various definitions of distributed systems have been given in the literature, none of them satisfactory and none of them in agreement with any of the others. For our purposes it is sufficient to give a loose characterization:

> *A distributed system is a collection of independent computers that appear to the users of the system as a single computer.*

This definition has two aspects. The first one deals with hardware: the machines are autonomous. The second one deals with software: the users think of the system as a single computer. Both are essential. We will come back to these points later in this chapter, after going over some background material on both the hardware and the software.

Rather than going further with definitions, it is probably more helpful to give several examples of distributed systems. As a first example, consider a network of workstations in a university or company department. In addition to each user's personal workstation, there might be a pool of processors in the machine room that are not assigned to specific users but are allocated dynamically as needed. Such a system might have a single file system, with all files accessible from all machines in the same way and using the same path name. Furthermore, when a user typed a command, the system could look for the best place to execute that command, possibly on the user's own workstation, possibly on an idle