Distributed Systems: Principles and Paradigms, Andrew Stuart Tanenbaum, Maarten Van Steen, Prentice Hall Pearson Education International, 2002, 0131217860, 9780131217867, . For courses on Distributed Systems, Distributed Operating Systems, and Advanced Operating Systems focusing on distributed systems found in departments of Computer Science, Computer Engineering and Electrical Engineering.Distributed systems are common. Computer scientists and engineers need to understand how the principles and paradigms underlying distributed systems software and be familiar with several real world examples. No other book systematically examines the underlying principles and how they are applied to a wide variety of distributed systems with the depth and clarity of this presentation.

DOWNLOAD FULL VERSION HERE


The ... International Conference on Distributed Computing Systems, 1989.

Distributed Operating Systems, Andrew S. Tanenbaum, Sep 1, 1995, 606 pages. As distributed computer systems become more pervasive, so does the need for understanding how their operating systems are designed and implemented. Andrew S. Tanenbaums.

Concurrent systems, Jean Bacon, Jan 1, 2002, Computers, 726 pages. Concurrency is at the heart of many topics within computer science and is the focus of this book from Jean Bacon. The successful straightforward approach, coupled with new.

Distributed Operating Systems And Algorithm Analysis, Chow, Sep 1, 2009, 550 pages.

Distributed Systems, Mrs.Shehal Kamalapur Mrs.Neeta Deshpande, Jan 1, 2009, 567 pages. Introduction to distributed systems, Examples of distributed systems, Characteristics, Goals, Hardware and software concepts, Design issues, Resource sharing and the web.


For courses on Distributed Systems, Distributed Operating Systems, and Advanced Operating Systems focusing on distributed systems found in departments of Computer Science, Computer Engineering and Electrical Engineering. Distributed systems are common. Computer scientists and engineers need to understand how the principles and paradigms underlying distributed systems software and be familiar with several real world examples. No other book systematically examines the underlying principles and how they are applied to a wide variety of distributed systems with the depth and clarity of this presentation.

First part of the book dedicates one chapter to each of seven key principles of all distributed systems - Communication, processes, naming, synchronization, consistency and replication, fault tolerance, and security, provides students with an understanding of the key principles, paradigms, and models on which all distributed systems are based.

This book started out as a revision of Distributed Operating Systems, but it was soon apparent that so much had changed since 1995, that a mere revision would not do the job. A whole new book was needed. Accordingly, this new book has a new title: Distributed Systems: Principles and Paradigms. This change reflects a shift in emphasis. While we still look at some operating systems issues, the book now addresses distributed systems in a broader sense as well. For example, the World Wide Web, which is arguably the biggest distributed system ever built, was not even mentioned in the original book because it is not an operating system. In this book it rates almost an entire chapter.

The book is structured in two parts: principles and paradigms. The first chapter is a general introduction to the subject. Then come seven chapters on individual principles we consider most important: communication, processes, naming, synchronization, consistency and replication, fault tolerance, and security.

Actual distributed systems are usually organized around some paradigm, such as "everything is a file." The next four chapters each deal with a different paradigm and describe several key systems that use that paradigm. The paradigms covered are object-based systems, distributed file systems, document-based systems, and coordination-based systems.

The book is intended for a senior-level or a graduate course in computer science. Consequently, it has a website with PowerPoint sheets and the figures used in the book in various formats. A manual with solutions to the exercises is available to professors using the book in a course. They should contact their Prentice Hall representative for a copy. Of course, the book is also well-suited for individuals outside of a university setting wishing to learn more about this important topic.

A number of people have contributed to this book in various ways. We would especially like to thank Arno Bakker, Gerco Ballintijn, Brent Callaghan, Scott Cannon, Sandra Cornelissen, Mike Dahlin, Mark Derbyshire, Guy Eddory, Amr el Abbadi, Vincent Freely Chandana Gamage, Ben Gras, Bob Gray, Michael van Hartskamp, Philip Homburg, Jeroen Ketema, Andrew Kitchen, Ladislav Kohout, Bob Kutter, Jussipekka Leiwo, Leah McTaggart, Eli Messenger, Donald Miller, Shivakant Mishra, Jim Mooney, Matt Mutka, Rob Pike, Krithi Ramamritham, Shmuel Rotenstreich, Sol Shatz, Gurdip Singly Aditya Shivram, Vladimir Sukonnik, Boleslaw Szymanski, Laurent Therond, and Leendert van Doom for reading parts of the manuscript and offering useful comments.

Finally, we would like to thank our families. Suzanne has been through this process an even dozen times now. Not once has she said: "Enough is enough" although surely the thought has occurred to her. Thank you. Barbara and Marvin now have a much better idea of what professors do for a living and know the difference between a good textbook and a bad one. They are now an inspiration to me.
to try to produce more good ones than bad ones (AST).

Marielle knew what she was in for when I told her I was in the book-writing business again. She has
been supportive from the start, noticing also that there was more fun and less frustration for me than
the last time ("Are you writing chapters only once this time?"). Having Elke on your lap at 6 o'clock in
the morning while writing is not such a good idea, but it kept me focused on correctly setting
priorities. In that respect, Max did a wonderful job as well, but being older than Elke, he also knew
when it was better to play with someone else. They are great kids (MvS).

Andrew Tanenbaum and Maarten van Steen cover the principles, advanced concepts, and
technologies of distributed systems in detail, including: communication, replication, fault tolerance,
and security. Intended for use in a senior/graduate level distributed systems course or by
professionals, this text systematically shows how distributed systems are designed and
implemented in real systems. Written in the superb writing style of other Tanenbaum books, the
material also features unique accessibility and a wide variety of real-world examples and case
studies, such as NFS v4, CORBA, DCOM, Jini, and the World Wide Web.

Most of the book is just waffle... he explains neither the general principles nor the implementation
specifics in great detail, but instead spends 10 pages explaining the obvious, follows that with a
page with good technical information, then a page of insightful commentary, but then continues
again with another 10 pages of pointless chatter.

Chapters 1 through 4 are a great introduction to Distributed Systems, in the case you have had less
than optimal training on the subject in the past - I read these chapters at the beginning of a recent
Distributed Systems graduate course since this was the situation I was in. Chapters 5 through 7,
which were the main concentration in the course, are also the heart of the text: Synchronization,
Consistency and Replication, and Fault Tolerance. The authors write very well, and the diagrams
are among the best I have seen, especially if you think visually like me. In my opinion, some of the
explanations are drawn out a bit much, or worded in a strange way, but this does not take away
from the text's substance. What does subtract from my high opinion of the book is the cover art,
which makes it look like a book one would read in grade school. At least one professor in the
graduate school I am attending is not interested in using the text for his DS courses for that very
reason.

He is best known as the author of MINIX, a free Unix-like operating system for teaching purposes,
and for his computer science textbooks, regarded as standard texts in the field. He regards his
teaching job as his most important work.[11] Since 2004 he has operated Electoral-vote.com, a
website dedicated to analysis of polling data in federal elections in the United States.

His books have been translated into many languages including Arabic, Basque, Bulgarian, Chinese,
Dutch, French, German, Greek, Hebrew, Hungarian, Italian, Japanese, Korean, Macedonian,
Mexican Spanish, Persian, Polish, Portuguese, Romanian, Russian, Serbian, Spanish.[17] They
have appeared in over 120 editions and are used at universities around the world.[18]

In the early 1990s, the Dutch government began setting up a number of thematically oriented
research schools that spanned multiple universities. These schools were intended to bring
professors and Ph.D. students from different Dutch (and later, foreign) universities together to help
them cooperate and enhance their research.

Tanenbaum was one of the cofounders and first Dean of the Advanced School for Computing and
Imaging (ASCI). This school initially consisted of nearly 200 faculty members and Ph.D. students
from the Vrije Universiteit, University of Amsterdam, Delft University of Technology, and Leiden
University working in the areas of advanced computer systems, especially parallel computing, and
image analysis and processing.

Tanenbaum remained Dean for 12 years, until 2005, when he was awarded an Academy
Professorship by the Royal Netherlands Academy of Arts and Sciences, at which time he became a
full-time research professor. ASCI has since grown to include researchers from nearly a dozen universities in The Netherlands, Belgium, and France. ASCI offers Ph.D. level courses, has an annual conference, and runs various workshops every year.

In 1987, Tanenbaum wrote a clone of UNIX, called MINIX (MiNi-uNIX), for the IBM PC. It was targeted at students and others who wanted to learn how an operating system worked. Consequently, he wrote a book[20] that listed the source code in an appendix and described it in detail in the text. The source code itself was available on a set of floppy disks. Within three months, a USENET newsgroup, comp.os.minix,[21] had sprung up with over 40,000 subscribers discussing and improving the system. One of these subscribers was a Finnish student named Linus Torvalds who began adding new features to MINIX and tailoring it to his own needs. On October 5, 1991, Torvalds announced his own (POSIX like) kernel, called Linux, which originally used the MINIX file system, but it is not based on MINIX code.[22]

Although MINIX and Linux have diverged, MINIX continues to be developed, now as a production system as well as an educational one.[23] The focus is on building a highly modular, reliable, and secure, operating system. The system is based on a microkernel, with only 5000 lines of code[24] running in kernel mode. The rest of the operating system runs as a number of independent processes in user mode, including processes for the file system, process manager, and each device driver. The system continuously monitors each of these processes, and when a failure is detected is often capable of automatically replacing the failed process without a reboot, without disturbing running programs, and without the user even noticing. MINIX 3, as the current version is called, is available under the BSD license for free.

In 2004, Tanenbaum created Electoral-vote.com, a web site analyzing opinion polls for the 2004 U.S. Presidential Election, using them to project the outcome in the Electoral College. He stated that he created the site as an American who "knows first hand what the world thinks of America and it is not a pretty picture at the moment. I want people to think of America as the land of freedom and democracy, not the land of arrogance and blind revenge. I want to be proud of America again."[30] The site provided a color-coded map, updated each day with projections for each state's electoral votes. Through most of the campaign period Tanenbaum kept his identity secret, referring to himself as "the Votemaster" and acknowledging only that he personally preferred John Kerry. A libertarian who supports the Democrats, he revealed his identity on November 1, 2004, the day prior to the election, also stating his reasons and qualifications for running the website.[30]

For the 2008 elections, he got every state right except for Indiana, which he said McCain would win by 2% (Obama won by 1%) and Missouri, which he said was too close to call (McCain won by 0.1%). He correctly predicted all the winners in the Senate except for Minnesota, where he predicted a 1% win by Norm Coleman over Al Franken. After 7 months of legal battling and recounts, Franken won by 312 votes (0.01%).

On October 7, 2011, Universitatea Petru Maior din Târgu Mureș (Petru Maior University of Târgu Mureș) granted Tanenbaum the Doctor Honoris Causa (honorary doctorate) title for his remarkable work in the field of computer science and achievements in education. The academic community is hereby honoring his devotion to teaching and research with this award. At the ceremony, the Chancellor, the Rector, the Dean of the Faculty of Sciences and Letters, and others all spoke about Tanenbaum and his work. The pro-rector then read the 'laudatio,' summarizing Tanenbaum's achievements. These include his work developing MINIX (the predecessor to Linux), the RFID Guardian, his work on Globe, Amoeba, and other systems, and his many books on computer science, which have been translated in many languages, including Romanian, and which are used at Petru Maior University.

On May 12, 2008, Tanenbaum received an honorary doctorate from Universitatea Politehnica din București. The award was given in the academic senate chamber, after which Tanenbaum gave a lecture on his vision of the future of the computer field. The degree was given in recognition of Tanenbaum's career work, which includes about 150 published papers, 18 books (which have been translated into over 20 languages), and the creation of a large body of open-source software,
including the Amsterdam Compiler Kit, Amoeba, Globe, and MINIX.

Andrew S. Tanenbaum has an S.B. degree from M.L.T. and a Ph.D. from the University of California at Berkeley. He is currently a Professor of Computer Science at the Vrije Universiteit in Amsterdam, The Netherlands, where he is head of the Computer Systems Department. He is also the Dean of the Advanced School for Computing and Imaging, an interuniversity graduate school doing research on advanced parallel, distributed, and imaging systems. Nevertheless, he is trying very hard to avoid turning into a bureaucrat.

In the past, he has done research on compilers, operating systems, networking, and local-area distributed systems. His current research focuses primarily on the design of wide-area distributed systems that scale to a billion users. This research is being done together with Dr. Maarten van Steen. Together, all his research projects have led to over 90 refereed papers in journals and conference proceedings and five books.

Prof. Tanenbaum has also produced a considerable volume of software. He was the principal architect of the Amsterdam Compiler Kit, a widely-used toolkit for writing portable compilers, as well as of MINIX, a small UNIX clone intended for use in student programming labs. Together with his Ph.D. students and programmers, he helped design the Amoeba distributed operating system, a high-performance microkernel-based distributed operating system. The MINIX and Amoeba systems are now available for free via the Internet.

Prof. Tanenbaum is a Fellow of the ACM, a Fellow of the IEEE, a member of the Royal Netherlands Academy of Arts and Sciences, winner of the 1994 ACM Karl V Karlstrom Outstanding Educator Award, and winner of the 1997 ACM/SIGCSE Award for Outstanding Contributions to Computer Science Education. He is also listed in Who's Who in the World. His home page on the World Wide Web can be found at URL http://www.cs.vu.nl/~ast/.

Maarten van Steen is currently an associate professor at the Vrije Universiteit, Amsterdam where he teaches operating systems, computer networks, and distributed systems. He has also given various highly successful courses on computer systems related subjects to ICT professionals from industry and governmental organizations.

Dr. van Steen studied Applied Mathematics at Twente University and received a Ph.D. from Leiden University in the field of software design techniques for concurrent systems. After his graduate studies he went to work for an industrial research laboratory where he eventually became head of the Computer Systems Group, concentrating on programming support for parallel applications.

After five years of struggling to simultaneously do research and management, he decided to return to academia, first as an assistant professor in Computer Science at the Erasmus University Rotterdam, and later as an assistant professor in Andrew Tanenbaum's group at the Vrije Universiteit in Amsterdam. Going back to university was the right decision; his wife thinks so too.

His current research concentrates on large-scale wide-area distributed systems, with an emphasis on locating mobile objects, system architecture, and adaptive distribution and replication. Together with prof. Tanenbaum he leads the Globe project in which a group of approximately a dozen researchers collaborate to develop a wide-area distributed system by the same name. The Globe system is described at http://www.cs.vu.nl/globe.

New! Introducing the tech.book(store), a hub for Software Developers and Architects, Networking Administrators, TPMs, and other technology professionals to find highly-rated and highly-relevant career resources. Shop books on programming and big data, or read this week's blog posts by authors and thought-leaders in the tech industry. > Shop now

Virtually every computing system today is part of a distributed system. Programmers, developers, and engineers need to understand the underlying principles and paradigms as well as the real-world application of those principles. Now, internationally renowned expert Andrew S. Tanenbaum...
with colleague Martin van Steen – presents a complete introduction that identifies the seven key principles of distributed systems, with extensive examples of each. Adds a completely new chapter on architecture to address the principle of organizing distributed systems. Provides extensive new material on peer-to-peer systems, grid computing and Web services, virtualization, and application-level multicasting. Updates material on clock synchronization, data-centric consistency, object-based distributed systems, and file systems and Web systems coordination. For all developers, software engineers, and architects who need an in-depth understanding of distributed systems.

His current research concentrates on large-scale distributed systems. Part of his research focuses on Web-based systems, in particular adaptive distribution and replication in (collaborative) content distribution networks. Another subject of extensive research is fully decentralized (gossip based) peer-to-peer systems for wired as well as wireless ad hoc networks.

I enjoyed and learned a lot from both of Tanenbaum's OS textbooks, but this is really awful. On the one hand, the descriptions of things such as RPC are so abstract that I can't see how anyone could be expected to understand what a real RPC system would look like; on the other hand, there's not nearly enough effort made to give a picture of how the systems discussed fit into the broader context of computer science, or relate to each other.

If you buy this book expecting to learn how to do some web, rmi, corba or any other kind of distributed systems development then don't buy this one. This book is now a good source of theoretical material, I'm currently using this book because of the theoretical material but often I have to complement the information with other books like "Distributed Systems: Concepts and Design (4th Edition)" (by: Coulouris) which has more in-depth RMI practices and is also a good information source.

While I am not a specialist in all topics described in this book, I found it to be imprecise and, occasionally, downright wrong or misleading in parts where I had specific knowledge to the contrary. Many times language would appear to be made purposefully ambiguous, as if the author did not quite know what he is talking about. This type of ambiguity may be fine in general literature, but a presumably scientific textbook talking about logical and structured discipline should not be so written.

If you're expecting a "how-to" manual on writing distributed systems, this isn't it. It has an excellent coverage of some fundamental principles - I used it as a text book for a distributed systems course and found it very useful. The course and this book changed the way I think about system architecture. Some readers may find the material dry - it is, but in the end it's rewarding. It helps to have an exam at the end to drive you through this book. To the reviewer who said that it doesn't mention what's wrong with distributed objects or NFS, you'll find that it gives you the tools to see past the glossy hype of whatever the latest fad is (distributed objects, web services, or whatever else) and ask serious questions about how it handles failure, security, replication, etc.

Having said that, there were glaring grammatical errors, especially towards the end (the chapter on Distributed File Systems). I am surprised that it got past the editors. Also I had to re read some sections several times before I understood them (like the part about reliable group communication). It's still better than going through individual papers, but a more readable revised edition wouldn't hurt.

Everyone is entitled to their opinion; and some of the more negative remarks on this book are without warrant. The book starts as an exposition into distributed computing and branches into a comprehensive 'overview' of both theory and technical implementations. If you're looking for a book that is specific to one particular region of distributed technology this isn't it. If you've a computer science background you should appreciate the work Tannenbaum and Steen put into this book. It's a great book for academia and reference by information technology professionals who desire to understand the fundamentals of distributed computing. But as some comments have alluded, this is book is but the beginning. As a final note, the material is well referenced so you can branch of into
the published works of others as needed.

I used this book for an online graduate class in which the instructor deferred all teaching, exercises, exams, and grading to the textbook. This is bad enough, but add to it that this textbook is a terrible read and a royal pain in the rear to reference and it's a miracle that I managed to eek out an A. I say miracle, but I really mean countless hours of research, sweat, and tears.

First, the reading is terribly wordy yet still amazingly vague. I read pages and pages of text which revealed precious little usable information. Then I'd read a barrage of facts condensed into a few heavy, indecipherable sentences that don't paint any sort of clear picture. There are a few attempts at humor sprinkled in which result in a head slap and a strong desire to throw the book directly into the garbage.

Second, the book is terribly organized. I'll give the authors the benefit of the doubt that organizing this book must be hard because every aspect depends on something else. The chapter layout (in a table of contents sense) is actually appropriate. However, the organization of information in each chapter is worse than terrible. There's a brief intro, but not really a overview of what they are going to discuss. It will then jump from point to point and back again and then to an unrelated point and then to a tangent and then back to the original point. Once you realize you've totally lost the thread they begin a new topic or a loosely-related case study. Trying to find any specific information (like, say, for an exercise question) is a lesson in futility. You never know where a speck of needed information will turn up, if at all.

Third, the exercises are ridiculous. They are poorly-worded, vague, and subjective. You will spend large amounts of time trying to determine what a question is really asking. As far as answering the questions, maybe 1 in 10 covers material substantially covered in the book. 2 in 10 cover some extension of what is contained in the book that you might piece together with critical thinking skills. 6 in 10 are wild tangents that may be mentioned in the chapter text but without any substantial treatment what-so-ever. Sometimes it might be covered in another chapter (previous or future chapters), but most of the time it requires extensive web research followed by a wild guess. The last 10% relate to a random sentence inserted into a seemingly random paragraph somewhere in the chapter.

Fictitious example: "Cows: Cows can be brown or black-and-white spotted. Cows have 8 stomachs. Cows are stinky. My sister has cows and she says they make lousy pets!" ... 30 pages later ... "Dogs: Dogs are mammals. Dogs like people. Dogs are primarily carnivores (unlike cows, who eat grass). Dogs are neat-o!" Question: "What might one typically find in a bovine's digestive tract?"

Now assume you've never heard of a cow or a dog. You'd look up bovine and determine that it's a cow. OK, let me go read about cows in the text. Well, it says they have 8 stomachs, that might be relevant. Let me search the internet, it says stomach acid, bile, feces, intestines, stomachs, throat, etc. Wait, there's something mentioned in the dog section about cows, is that relevant? Hmm, let me throw all of this into some giant abomination of an answer. Oops, I missed points because I didn't think about all the bacteria in the colon. I'll try harder next time.

3) don't skim the chapter looking for relevant information - actually read the entire 80-130 page chapter. While doing so, write down all the bold words and definitions because they are not collected for you at the end of the chapter. Also, answer any questions you can along the way. If something sounds relevant to the question, but doesn't answer it, then note the page and keep reading. It just might surface again later. Don't let any sentence go unread, no matter how irrelevant the paragraph sounds, because it just might be the one you need.

5) do some critical thinking and then make a bunch of wild guesses to finish your assigned exercises. Make sure to cover every possible interpretation of the question. Explain both sides of the argument even if they only ask for one side. Pray for partial credit or that your instructor doesn't care enough to actually read your answers.
I gave this book 2 stars because it really does contain a lot of material and concepts regarding Distributed Systems. I gave it only 2 stars because it is painful to extract this information and get it into any sort of useful representation in your head. It is worthless as a reference and it fails as a textbook. If you have to use it in class, I feel bad for you. If you have to use it in class and your instructor is worthless, then I would recommend dropping the class. If you can't, then may God have mercy on your soul.

This work is protected by local and international copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from this site should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.

Very few textbooks today explore distributed systems in a manner appropriate for university students. In this unique text, esteemed authors Tanenbaum and van Steen provide full coverage of the field in a systematic way that can be readily used for teaching. No other text examines the underlying principles and their applications to a wide variety of practical distributed systems with this level of depth and clarity.

Nobody is smarter than you when it comes to reaching your students. You know how to convey knowledge in a way that is relevant and relatable to your class. It's the reason you always get the best out of them. And when it comes to planning your curriculum, you know which course materials express the information in the way that's most consistent with your teaching. That's why we give you the option to personalize your course material using just the Pearson content you select. Take only the most applicable parts of your favorite materials and combine them in any order you want. You can even integrate your own writing if you wish. It's fast, it's easy and fewer course materials help minimize costs for your students.

Our library is vast, and it's all at your fingertips. Create a custom book by selecting content from any of our course-specific collections. Here, you'll find chapters from Pearson titles, carefully-selected third-party content with copyright clearance, and pedagogy. Once you're satisfied with your customized book, you will have a print-on-demand book that can be purchased by students in the same way they purchase other course material.

Pearson Learning Solutions will partner with you to select or create eBooks, custom eBooks, online learning courses, resource materials, teaching content, media resources and media supplements. Simply share your course goals with our world-class experts, and they will offer you a selection of outstanding, up-to-the-minute solutions.

Pearson Learning Solutions offers a broad range of courses and custom solutions for web-enhanced, blended and online learning. Our course content is developed by a team of respected subject matter experts and experienced eLearning instructional designers. All course content is designed around specific learning objectives.

Andrew Stuart "Andy" Tanenbaum (sometimes referred to by the handle ast)[1] (born March 16, 1944) is a professor of computer science at the Vrije Universiteit, Amsterdam in the Netherlands. He is best known as the author of MINIX, a free Unix-like operating system for teaching purposes, and for his computer science textbooks, regarded as standard texts in the field. He regards his teaching job as his most important work.[2]

Tanenbaum was born in New York City and grew up in suburban White Plains, New York. He received his B.Sc. degree in Physics from MIT in 1965. He received his Ph.D. degree in physics from the University of California, Berkeley in 1971. He moved to the Netherlands to live with his wife, who is Dutch, but he retains his United States citizenship. He teaches courses about Computer Organization and Operating Systems and supervises the work of Ph.D. candidates at the VU
University Amsterdam.

Description: Presents a complete introduction to distributed principles and paradigms. Author identifies the seven key principles of distributed systems, and presents extensive examples of each. For all developers, software engineers, and architects who need an in-depth understanding of distributed systems.