

## Q&A

# Systematic Thinking

*Andrew S. Tanenbaum talks about MINIX, microkernels, and electronic voting systems.*

ANDREW S. TANENBAUM, a professor of computer science at the Vrije Universiteit in Amsterdam, has been at the forefront of operating systems design for more than 20 years. For an appreciation of Tanenbaum's sense of humor, academic musings, and philosophy of life, see his homepage's FAQ (<http://www.cs.vu.nl/~ast/home/faq.html>).

**You are best known for MINIX, the Unix-like operating system that you created in 1987.**

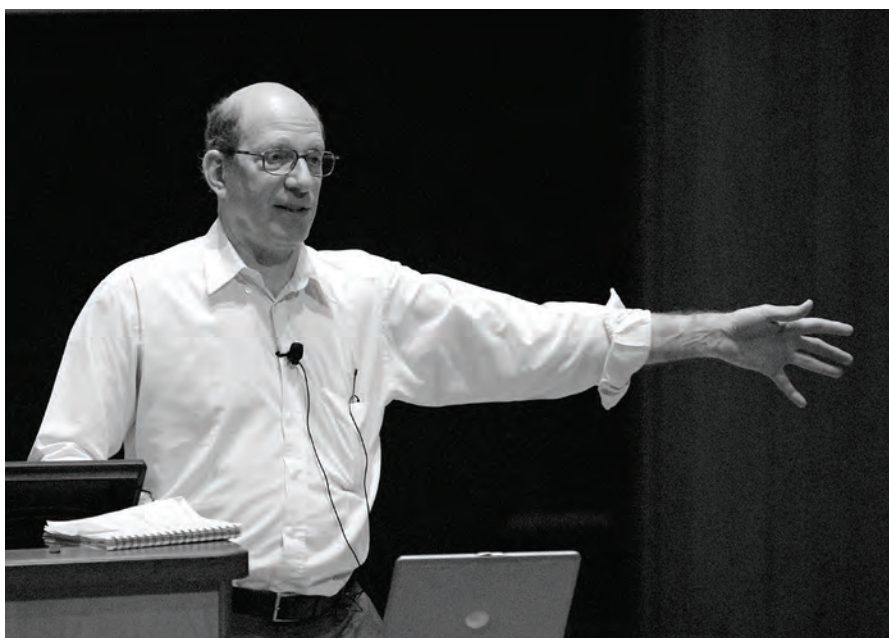
Version 6 of UNIX was widely used in universities, and it was a popular tool in operating systems courses. Then some bean counter at AT&T said, "Gee, if we keep this secret, we can make more money," so they put a clause in the version 7 contract saying you couldn't teach it anymore. At that point I decided I would write something that looked quite a bit like UNIX but was my own code and wasn't tied to their license.

**What's going on with MINIX now? I understand you continued to develop and refine it over the years.**

Yes, and in 2004 I decided to pick it up again and really make the point that I think microkernels are a more reliable way of doing things.

**This is the idea that we can build more reliable operating systems by breaking up the components.**

I find it very peculiar that anybody believes you can take anything as complicated as an operating system and have it run as one gigantic program. It's just too complicated. I mean, there are



Andrew S. Tanenbaum giving a keynote presentation at linux.conf.au 2007.

other complicated things in the world, but they're generally modular.

**And that's why you decided to base MINIX on a microkernel rather than a monolithic kernel.**

Yes, a microkernel is a very small piece of code running in kernel mode that provides the basis for the system, and doesn't do very much itself. Then the operating system runs as a collection of other processes in user mode. So, instead of one big program that does everything, you've got one program that drives the disk, one program that handles the audio, and one program that manages memory, each doing one, specific task and communicating with the others by well-defined protocols.

**So by breaking up the operating system into well-defined pieces, you make it more reliable.**

It also means you can replace pieces independently, which has a variety of consequences, such as security. MINIX has all these components—these servers and device drivers that run the user processes. Each piece has certain powers. And there's a data structure within the kernel that tells, for every process, exactly what it's allowed to do. So if a hacker found a vulnerability in the audio driver, took it over, and wanted to fork it, the kernel would say, "Sorry, you don't have permission to fork, there's no reason for audio drivers to do that." It's not perfect, of course, [CONTINUED ON P. 111]

[CONTINUED FROM P. 112] but it puts a lot of obstacles in the way of somebody trying to get into the system.

### How do you think microkernel systems can penetrate the broader consumer marketplace?

I don't think it's going to be easy, but I can think of a couple routes by which it could happen. Some people in the European Union have talked about changing regulations to require software to fall under the same liability laws as everything else. If you make a tire, and one in 10 million explodes, you can't say, "Well, tires explode sometimes." Why isn't software like any other product? Imagine if there were some liability associated with its not working. Manufacturers would suddenly be very interested in making things reliable.

### You've also worked on security-related projects, like an e-voting system.

The trouble with voting machines is knowing that the software can be trusted. And cryptographers who work in this area have complicated schemes

**"I think microkernels are a more reliable way of doing things."**

to address that, but the schemes are so complicated that virtually nobody except a professional cryptographer could understand them. We've designed a scheme which, though it uses cryptography, is much simpler and easier to use.

We assume that the software is open source. At the time you go to vote, you could come with some handheld device and query the voting machine, "Give me a cryptographic checksum of the software currently in your memory." Then you could check that and know the software running on the machine is the software that's supposed to be there. We have a lot of other design issues dealing with how the keys

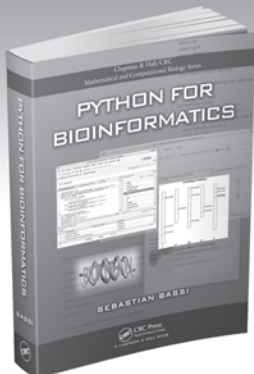
are distributed, and not trusting any single party. But basically we're trying to design a system that's more secure than current ones and less subject to hacking.

### It sounds like a very comprehensive approach.

I've always tried to look at the systems aspect of a problem. I've also done work on sensor networks—people have proposed dropping sensors along the national border to prevent people from sneaking in. The only thing they worry about is, "Suppose someone captures a sensor and steals all the keys?" What they hadn't considered is, "What is the range of a sensor? How does it detect the difference between an illegal immigrant and a rabbit?" Any system will be attacked at the weakest link, so you have to pay attention to where the weakest link is, and not which is the most mathematically interesting problem. □

Leah Hoffmann is a Brooklyn, NY-based technology writer.

© 2010 ACM 0001-0782/10/0400 \$10.00



### Python for Bioinformatics

Sebastian Bassi  
Universidad Nacional de Quilmes, Bernal, Argentina  
Requiring no prior knowledge of programming-related concepts, this volume focuses on the easy-to-use, yet powerful, Python computer language. It contains working code that solves real-world biological problems and includes a DVD with a ready-to-run virtual machine (VM) based in DNALinux to test the code.

Catalog no. C9292  
January 2010, 587 pp., Soft Cover  
ISBN: 978-1-58488-929-8, \$69.95 / £44.99  
After Savings: \$55.96 / £35.99

**Save 20% on these new titles in data mining, bioinformatics, and machine learning from CRC Press!**



**CRC Press**  
Taylor & Francis Group

### Coming Soon!

#### Patterns of Data Modeling

Michael Blaha  
OMT Associates, Inc. Placida, Florida, USA  
"... The breadth of coverage is enormous, ranging from basic data structures through star schema, archetypes for representing commonly found concepts, and canonical models for tough problems."

—Paul C. Brown, Principal Software Architect, TIBCO Software Inc.

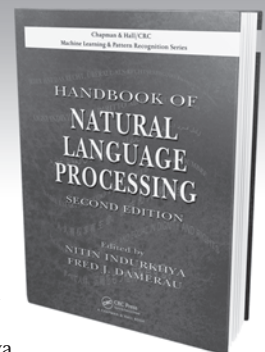
Catalog no. K11048  
April 2010, c. 264 pp., Soft Cover  
ISBN: 978-1-4398-1989-0, \$49.95 / £31.99  
After Savings: \$39.96 / £25.59

#### Handbook of Natural Language Processing

Second Edition  
Edited by  
Nitin Indurkha  
University of New South Wales, Sydney, Australia  
Fred J. Damerau  
Goshen, Connecticut, USA

This comprehensive, modern handbook presents practical tools and techniques for implementing natural language processing in computer systems. It includes a new applications section, a broader multilingual scope to include Asian and European languages, and an actively maintained wiki (<http://handbookofnlp.cse.unsw.edu.au>) that provides supplementary information.

Catalog no. C5921  
February 2010, c. 704 pp.  
ISBN: 978-1-4200-8592-1, \$99.95 / £63.99  
After Savings: \$79.96 / £51.19



Enter promo code 694AM at [www.crcpress.com](http://www.crcpress.com) to receive 20% discount.

Offer expires May 31, 2010.