Lessons Learned from 30 Years of MINIX?

Andrew S. Tanenbaum
Vrije Universiteit
PREHISTORY

• MINIX wasn’t created in a vacuum
• There was a lot of history before it
• Let’s review some of it
• In 1960, M.I.T. had an IBM 7094
• It filled a big room and ran batch jobs from cards
IBM 7094
COMPARISON OF IBM 7094 AND iPad

<table>
<thead>
<tr>
<th>Item</th>
<th>IBM 7094</th>
<th>iPad Air</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>0.4 MIPS</td>
<td>2 x 1400 MIPS</td>
<td>7,000x</td>
</tr>
<tr>
<td>Memory</td>
<td>0.288 MB</td>
<td>1024 MB</td>
<td>3,500x</td>
</tr>
<tr>
<td>Ext storage</td>
<td>3MB</td>
<td>128 GB</td>
<td>800x</td>
</tr>
<tr>
<td>Volume</td>
<td>800 m$^3$</td>
<td>0.0004m$^3$</td>
<td>2,000,000x</td>
</tr>
<tr>
<td>Price (2016 $)</td>
<td>$12 million</td>
<td>$900</td>
<td>13,000x</td>
</tr>
</tbody>
</table>

Multiplied out: $5 \times 10^{20}$ x better
**IF BOEING 707 HAD THIS IMPROVEMENT**

<table>
<thead>
<tr>
<th>iPad Item</th>
<th>Aircraft item</th>
<th>Aircraft could</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>Flight time</td>
<td>AMS to SFO in 12 sec</td>
</tr>
<tr>
<td>Memory</td>
<td>Passengers</td>
<td>700,000 passengers</td>
</tr>
<tr>
<td>Disk space</td>
<td>Range</td>
<td>160x around earth nonstop</td>
</tr>
<tr>
<td>Volume</td>
<td>Size</td>
<td>Size of iPhone</td>
</tr>
<tr>
<td>Price (2016 $)</td>
<td>Price</td>
<td>Round-trip ticket for 9 cents</td>
</tr>
</tbody>
</table>

- But your baggage would still end up in Siberia
- One in 50 flights would crash
- The engineers would be proud of this safety record
CTSS

- Jobs were turned in as a deck of punched cards
- It took hours to get a job back
- If you entered one comma wrong, job failed
- Very hard to get work done
- In 1961 to improve matters, M.I.T created CTSS
- This was an online system, response of seconds
- Huge success
MULTICS

- Successor to CTSS was MULTICS
- MULTIplexed Information & Computing Service
- Designed in 1964
- M.I.T. partnered with General Electric & Bell Labs
- MULTICS had a troubled youth
- When first compiled, the kernel didn’t fit in RAM
- In 1969 Bell Labs dropped out
- A programmer, Ken Thompson went back to NJ
UNICS

- Ken found an old PDP-7 at Bell Labs and rewrote it
- It could support only one user
- Brian: UNIplexed Information & Computing Service
- For short: UNICS
- Puns about EUNUCHS being castrated MULTICS
- Spelling later changed to UNIX
UNIX AT BELL LABS

• When the PDP-11 came out Bell Labs bought one
• Dennis Ritchie invented C and wrote the compiler
• Ken & Dennis rewrote Ken’s system for the PDP-11
• Went through 6 editions of the manual internally
• Universities wanted it, Bell Labs agreed
• V6 was licensed in 1976
• John Lions wrote a book describing it line by line
• Bell Labs lawyers didn’t like this at all
• V7 (1979) came with a license that said: no books
MINIX

• In 1984 I decided to rewrite V7 for the IBM PC
• They cost about $1500
• Possible for a student to own one
• It took me 2 years, evenings and weekends
• It sort of worked and I could use it for development
• I was very conscious of flaws in the system
• Lesson: Eat your own dog food
IT CRASHED AFTER AN HOUR

- I didn’t understand why it crashed after an hour
- In desperation, I wrote a PC simulator
- It worked perfectly on the simulator!
- I told my student, Robbert van Renesse
- He said: I heard 8088 gives interrupt 15 when hot
- I said: Nothing about that in the documentation
- I changed it to catch interrupt 15 and it worked
- **Lesson: Don’t trust the documentation**
- **Lesson: Your students might know more than you**
I WROTE A BOOK ABOUT MINIX

- In 1987 I published a book describing MINIX
- Like Lions book, but no hassle with Bell Labs
HOW TO DISTRIBUTE THE SOFTWARE?

• Almost no one had an Internet connection then
• Publisher sold box of 8 floppies + 500 pg book for $69
• Lesson: You need a way to distribute your product
POST-RELEASE REACTION

• Bookstore owner in California asks me to visit
• I figured he would set up a table & I’d sign books
• He rented the Santa Clara Convention Center
• And filled it up
• USENET Newsgroup had 40,000 readers in 1 mo.
• People started contributing software
TEACHING VS. FULL-BLOWN OS

• Many people wanted full-blown OS
• I was afraid it would get to complicated to study
• Also, I thought BSD or GNU would happen
• I resisted trying to make it into full-blown OS
• Nevertheless a lot of good software came in
• Jan-Mark Wams wrote a very good test suite and a compression program that saved 2 floppy disks
• Later important for people with 2400 bps modems
• **Lesson: Size matters**
32-BIT PROCESSOR ARRIVES

• In 1985, Intel released the 32-bit 386
• Bruce Evans ported MINIX to 386
• Code distinguished kernel mode from user mode
• Code distinguished physical & virtual addresses
• Neither were actually necessary on 8088
• Doing so made the port to the 386 much easier

• Lesson: Think about future hardware
OLIVETTI PC

- MINIX on Olivetti PC acted strangely
- As if the keyboard were broken
- All Olivettis had the same problem
- Discovery: Italian computers have different keybd
- Input code was changed to accept national keybds
- Lesson: If someone gives you a lemon, make lemonade
• In 1991 Linus Torvalds bought a PC to run MINIX
• On 29 March 1991 he posted to comp.os.minix
• 10 Days later he mocked a another newbie
• Clearly as an educational tool, MINIX succeeded
• In Aug. 1991, Linus said: I’m writing a new OS
• He developed it using MINIX
• This eventually became the Linux kernel
• In 1992, I said microkernels were better than monolithic designs
• Big flamewar with Linus Torvalds ensues
• 24 years later I still get lots of mail about this
• Lesson: The Internet is like an elephant; it never forgets
MORE ON MICROKERNELS

• Windows NT was originally a microkernel design
• Performance forced everything back into kernel
• OS X is FreeBSD on top on Mach microkernel
• L4 microkernel is inside a billion cell phones
• QNX: widely used microkernel in embedded systems
IS TODAY’S SOFTWARE FOREVER?

- Linux is written in gcc
- MINIX is written in ANSI Standard C
- What happens when a better compiler comes out?
- MINIX switched easily to Clang/LLVM
- Linux is still stuck with gcc
- **Lesson:** Stick with international standards
AT&T VS. BSDI

• BSD developers formed a company to sell BSD
• AT&T sued them
• This handcuffed BSD from 1992 to 1994
• This gave Linux a window with no BSD to fight
• Eventually settled out of court
• AT&T still didn’t know what to do with UNIX
• Lesson: When a tiny startup appears that knows more than your company, buy them
MINIX V2

- In 1997 MINIX 2 came out
- Was POSIX compatible instead of V7
- In 2000 I convinced my publisher to release it under BSD license and put it on the Internet
- I should have done this much earlier
- **Lesson: Reexamine your plan periodically**
ACADEMY PROFESSOR

- In 2004, I got a grant to do research on reliability
- In 2004, I became an Academy Professor
- This gave me €2 million to use MINIX for research
- Lesson: Doing something outside the mainstream can get you research funding if it is important
MOVING DRIVERS OUT OF THE KERNEL

• Initially device drivers were in kernel address space
• But they were scheduled as processes
• Because context switch was expensive on 8088
• Jorrit Herder moved them all to user space
• Failed drivers could be replace on the fly
• Now we had something no one else had
• Lesson: Each driver should run as a user process
• Lesson: Try for an early success; it builds morale
MINIX 3

- In 2005, we released MINIX 3
- This was a much more serious system
- MINIX 1 was unfortunately very well known
- No one believed MINIX 3 was really different
- Windows 95/98 were just MS-DOS with a GUI
- But Microsoft renamed it “Windows” – smart!
- **Lesson: If V3 != V1, give it a new name**
EUROPEAN RESEARCH COUNCIL GRANT

- EU has been thinking about product liability for SW
- Suppose 1 in 10 million tires explodes
- Manufacturer can’t say: Tire explosions happen”
- For software that works
- In 2008 I got a €2.5 grant for OS research
- Goal was to make MINIX into reliable system
NEW RESEARCH GROUP

• I could hire
  – 4 expert programmers
  – 6 Ph.D. students & a postdoc
• Programmers built a very solid product
• Students did research
• Everybody had their own copy of the code
• We used source code control systems, git, etc.
• Still we had huge problems merging code later
• **Lesson:** Combining research & a product is hard
HOW TO GET USERS

• We discovered that MINIX couldn’t do anything
• So we made it NetBSD compatible
• This gave us 6,000 packages for free
• Lesson: To be used, a system must do something
EMBEDDED SYSTEMS

• We also decided to focus on embedded systems
• We ported MINIX to the ARM
• The port was straightforward
• There was no inline x86 code ever in MINIX
• Lesson: Today’s hardware may not be tomorrow’s
MINIX 3.4 RELEASE CANDIDATE 2

• Main new feature: live update
• Goal: no more reboots
• Can be used for security fixes or new versions
• Replace drivers and servers without rebooting
• Summary:
  – Start a new driver or server as a new process
  – Transfer the old state to the new process
  – Cut over to the new one
• Lesson: It is hard to change existing ways of doing things
• In this case, microkernels
• Other examples
  – FORTRAN
  – Windows XP (still 250 million XP machines running)
  – QWERTY keyboard
  – Interlaced NTSC television sets
  – Magnetic stripe credit cards
• I wrote a paper covering much of this material
• It will be in the March 2016 Communications of the ACM
MASTERS PROGRAM AT THE VU

• PDCS: Parallel and Distributed Computer Systems
• Focus is on experimental computer science
• Design, implementation, and testing of software
• Many lab courses, little theory
• Emphasis on research
• Opportunity to be a serf
• Good preparation for a Ph.D. later
• See pdcs.vu.nl
THE END
The MINIXCon 2016 program is now posted. To see it and register, click here

What Is MINIX 3?

MINIX 3 is a free, open-source, operating system designed to be highly reliable, flexible, and secure. It is based on a tiny microkernel running in kernel mode with the rest of the operating system running as a number of isolated, protected, processes in user mode. It runs on x86 and ARM CPUs, is compatible with NetBSD, and runs thousands of NetBSD packages. Get MINIX 3 now and join our community!

- Read More
- Download

LEARN MORE ABOUT MINIX 3
- FEATURES. See a list of the MINIX 3 features
- FAQ. Get quick answers to common questions
- RESOURCES FOR NEWBIES. Info for beginners
- DOCUMENTATION. Various articles
- NEWS. Read our new posts
- RESEARCH. MINIX 3 as a base for research

GET INVOLVED WITH MINIX 3
- GETTING STARTED. If you are new to MINIX 3
- DOWNLOAD. Get the current version as a CD-ROM image
- NEWSGROUP. Ask question and get answers
- HELP US. How can I help the project?
- HIRE A CONSULTANT. Tailor MINIX 3 to your company's needs
- DONATE. All financial contributions are welcome

Get a masters degree in Computer Systems at the university where MINIX 3 was written, VU University, in Amsterdam.
DISCUSSION – THE FUTURE OF MINIX

• How to build a community?
• Steering committee: rules for selecting members? tasks?
• What niche is there for MINIX3?
• What roadmap is needed to fill that niche?
  – Other platforms? Raspberry Pi?
  – Key software missing that people need?
• How to improve collaboration?
  – Conference in 2017? For €100?
• What kind of PR do we need?
• Better documentation on the wiki?
• As an open-source project, we need serious volunteers
• Note: Continue discussion later on the newsgroup