CSCI 1670/2670
Operating Systems

Staff

- Professors
  - Tom Doeppner
  - Malte Schwarzkopf
- Head TAs
  - Ayman Benjelloun Touimi
  - Matthias Yee
- UTAs
  - Angel Arrazola
  - Alexander Cueva
  - Tomas Dougan
  - Brandon Gong
  - Mithi Jethwa
  - Mason Lee
  - Daniel Liu
  - Jason Ni
  - Nicholas Yarnall
  - Kyle Yeh

The Course

- Implement threads
- Learn about operating systems
- Implement one (Weenix)
  - file-system-related components (1670)
  - most of the rest of it (1690, 2670)

Workload Components

- Four moderate programming assignments
  - UThreads: simple user-level threads package
  - MThreads: more sophisticated user-level threads package
  - VFS: system calls and high-level file system
  - S5FS: low-level file system
- One large programming assignment: Weenix
  - combines VFS and S5FS, as well as Processes, Drivers, and Virtual Memory
- Four homeworks
- In-class TopHat questions

Workload by Course

- 1670 Section 1
  - quizzes: 10%
  - homeworks: 40%
  - moderate programming assignments: 50%
- 1670 Section 2
  - homeworks: 44.44%
  - moderate programming assignments: 55.56%
- 1690
  - large programming assignment: 100%

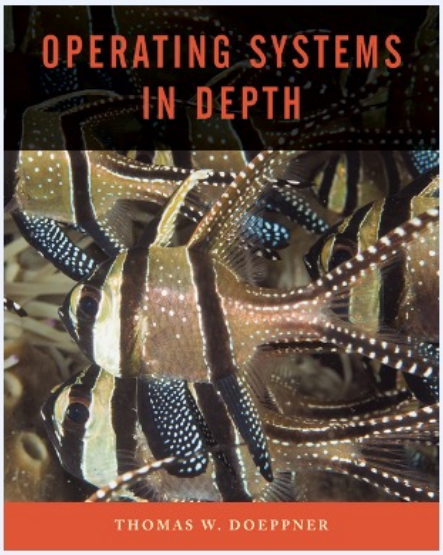
Workload by Course

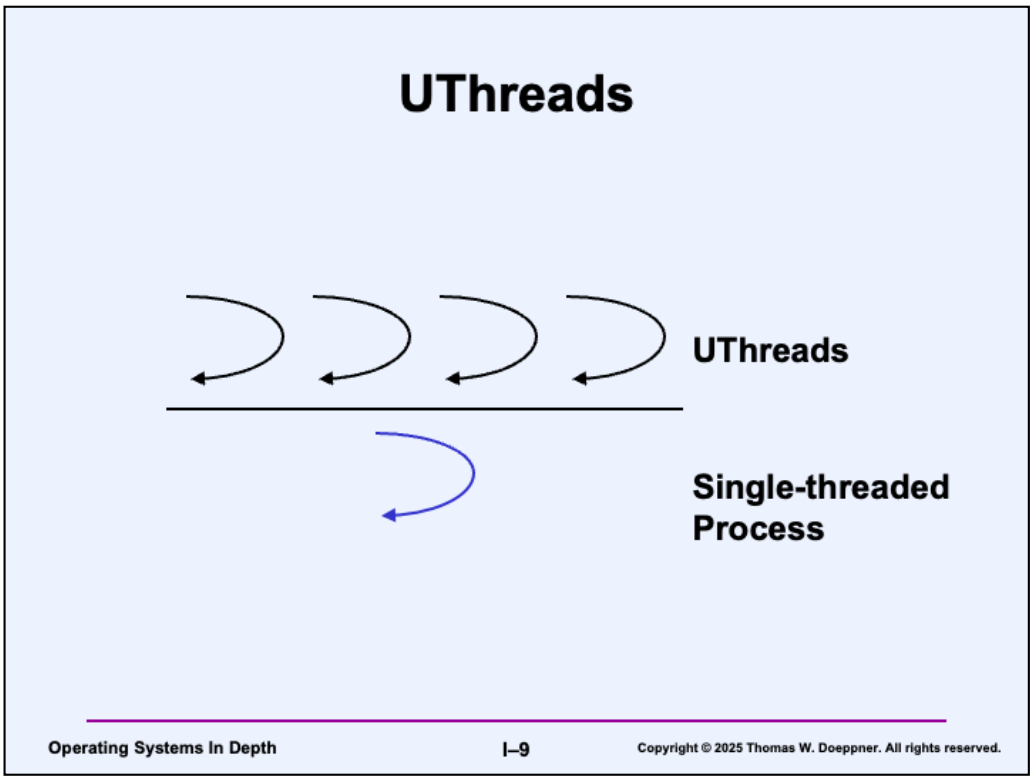
- 2670 Section 1
  - quizzes: 6.67%
  - homeworks: 26.68%
  - moderate programming assignments: 33.32%
  - large programming assignment: 33.33%
- 2670 Section 2
  - homeworks: 28.58%
  - moderate programming assignments: 35.71%
  - large programming assignment: 35.71%

Skills Needed

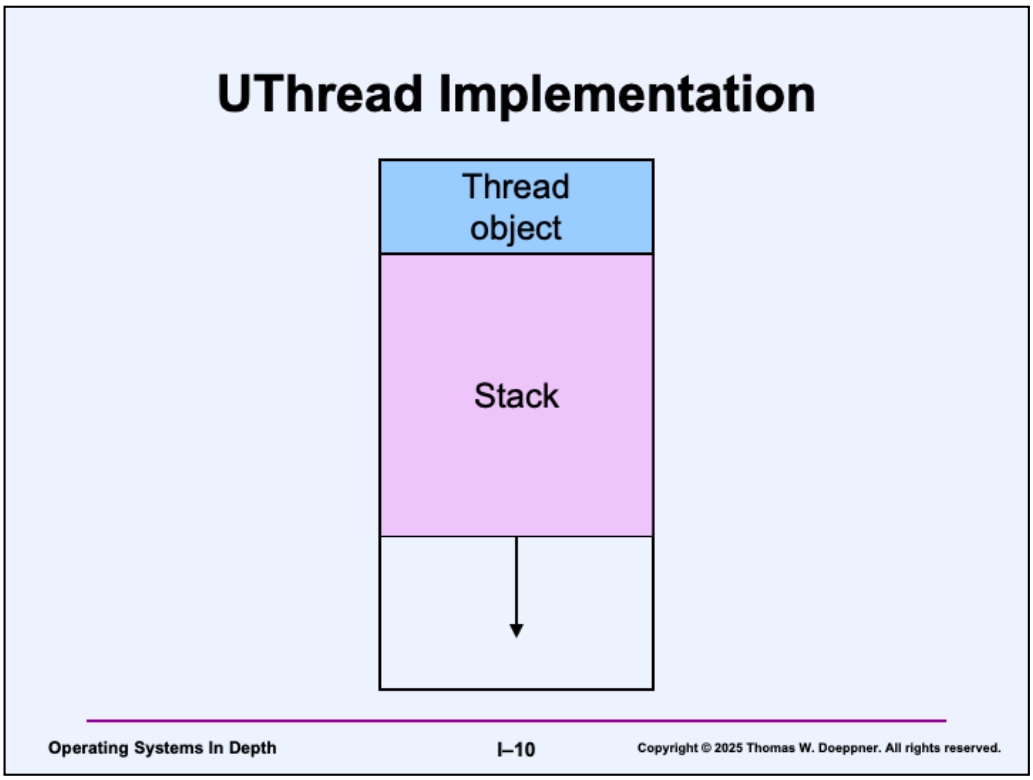
- Ability to write and debug largish programs in C with POSIX threads
  - CSCI 330/1330 or CSCI 300/1310 (you may want to review POSIX threads and signals material from 330 lectures)
- Basic computer architecture
  - CSCI 330/1330 or CSCI 300/1310

Textbook

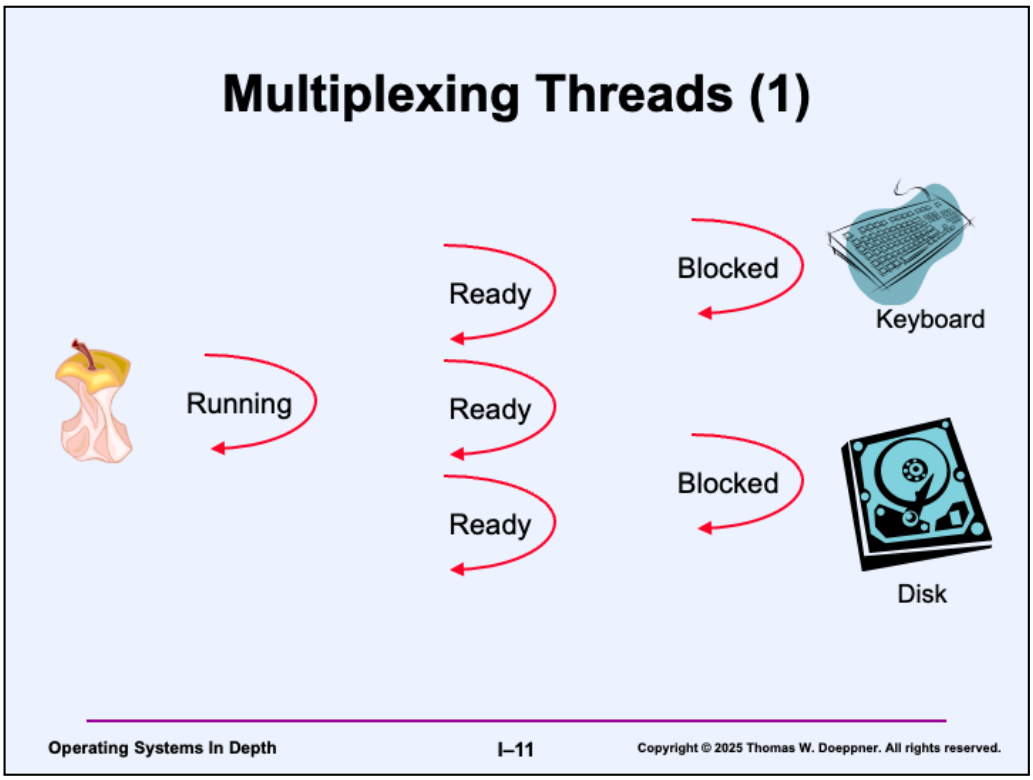




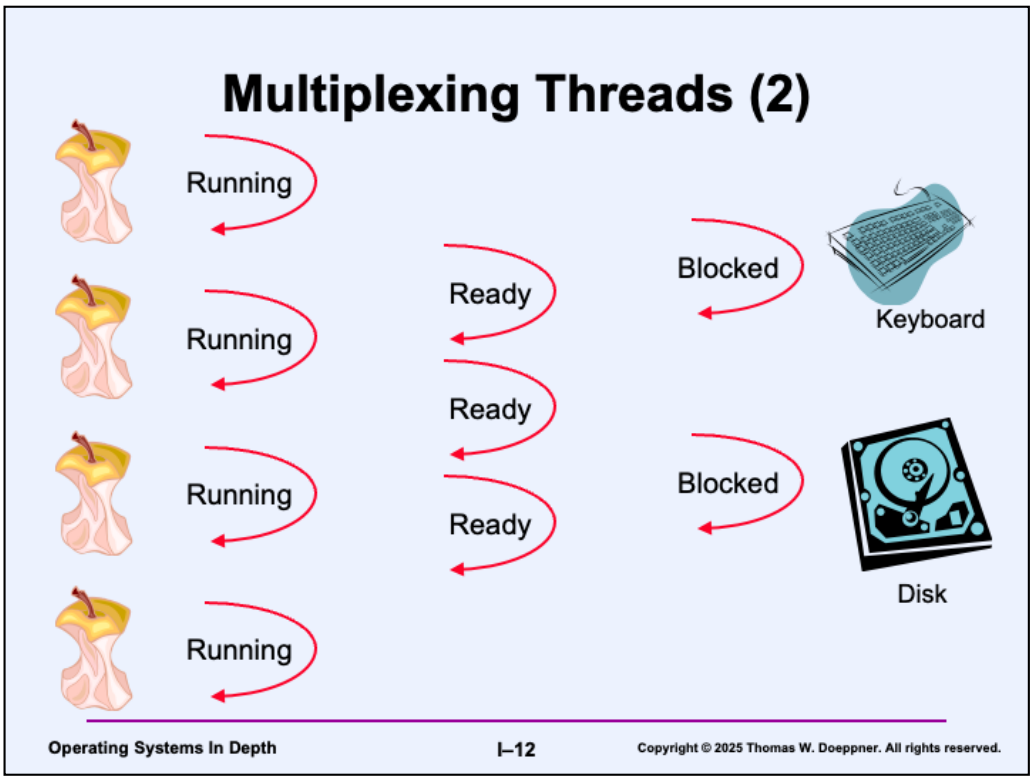
Our first project is to implement a subset of POSIX threads. The OS (Linux) will support a single thread in your process. Your code will multiplex multiple Uthreads on this single Linux thread.



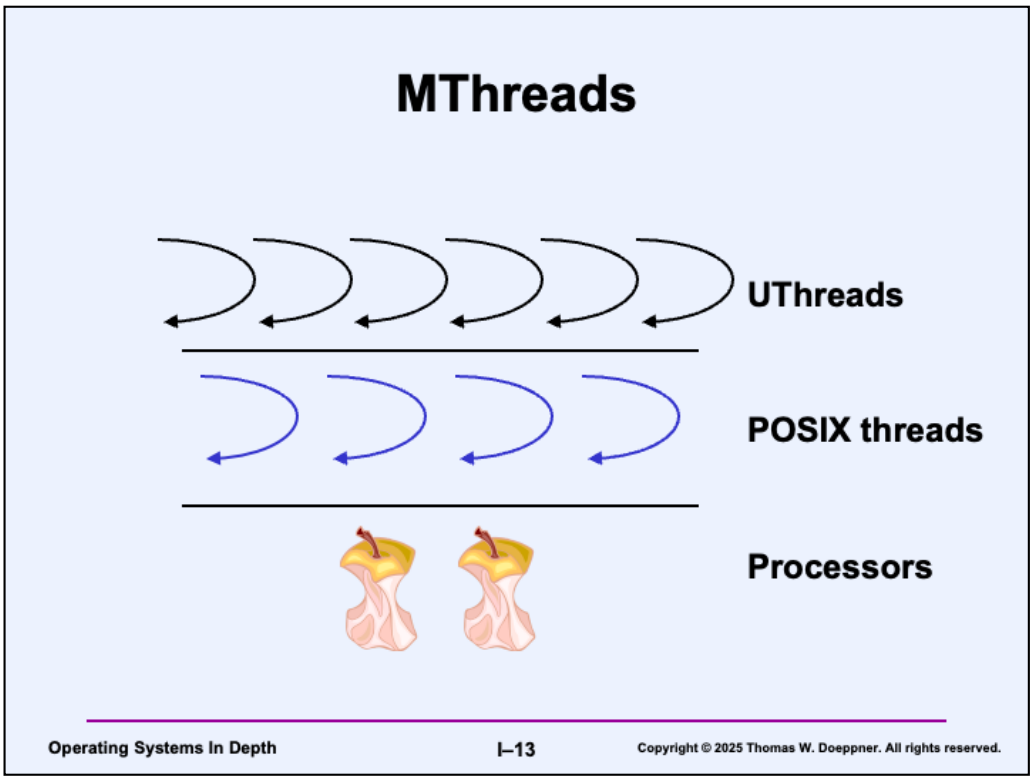
Whenever a uthread is created, your code will allocate a stack and a thread object to represent it. The thread object keeps track of the thread's stack pointer and instruction pointer, and represents the thread in queues.



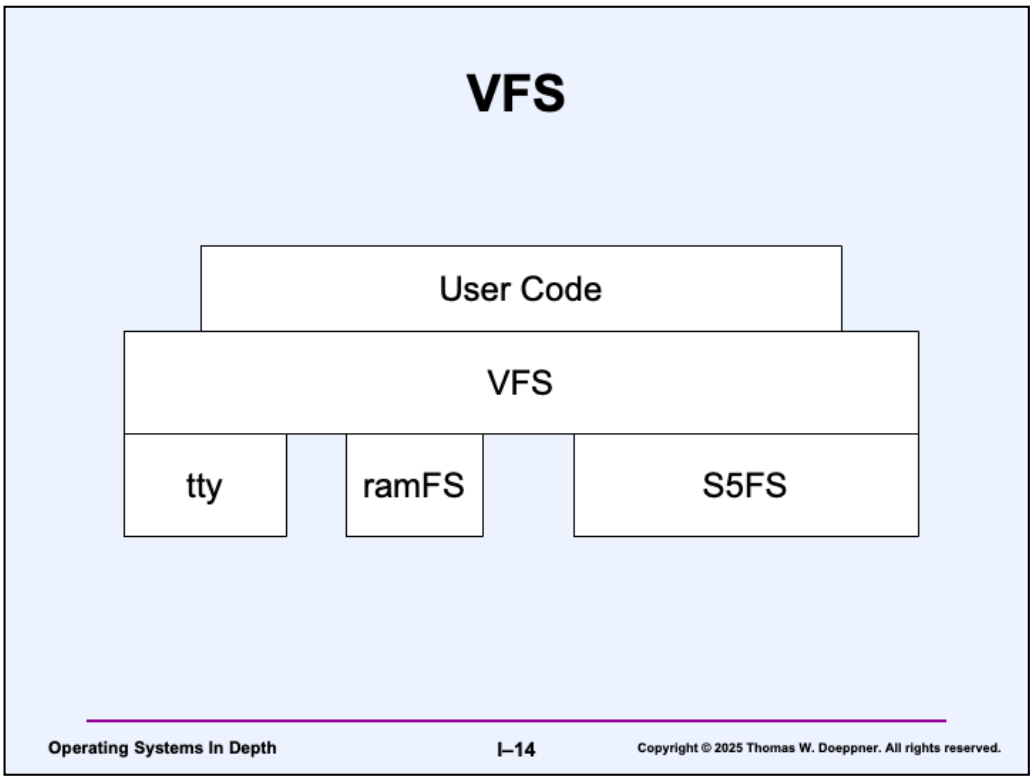
We can think of the Linux thread (that we're multiplexing) as representing the processor, which we're sharing among all the uthreads. As we discussed in CS33, some of the uthreads would be ready to run, though not currently assigned to the processor. These we call the **ready** threads. Other threads are **blocked**, waiting for something to happen (such as a mutex being unlocked, an I/O request to complete, etc.).



A more complicated situation is when we have more than one processor, and we'd like the uthreads to be multiplexed over all the processors. Thus, a ready thread might run on any available processor. However, our uthreads implementation doesn't make this possible, since the uthreads are multiplexed on only a single Linux thread.



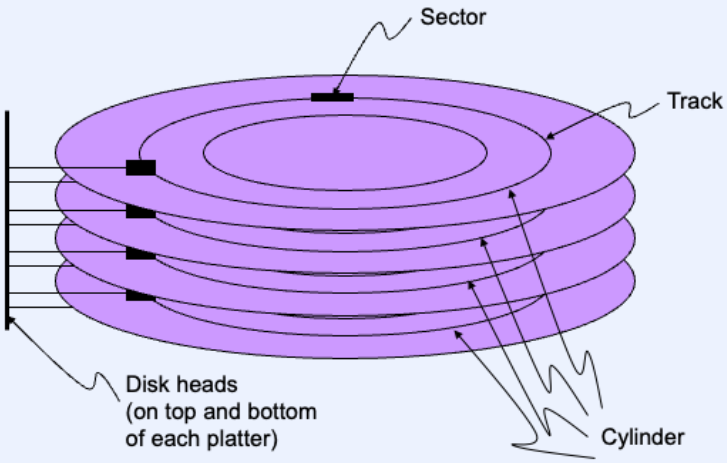
In the MThreads project, we extend our UThreads implementation by multiplexing them on multiple (OS-supported) Linux threads. Thus from your (the implementor's) perspective, the POSIX threads are processors and you are implementing UThreads using a system with multiple processors. This results in many more synchronization issues than does UThreads and provides a good introduction to how an OS is actually implemented.



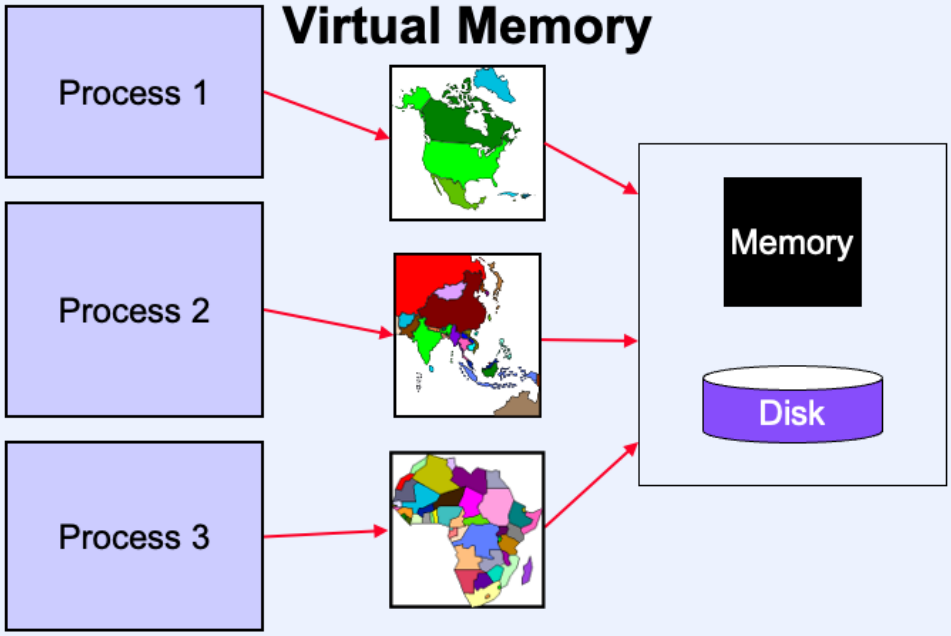
In principle, there could be other things below the VFS layer, such as socket-level communication protocols, distributed file systems, etc. But what's show in the slide is what we'll be working with in our Weenix project.

S5FS

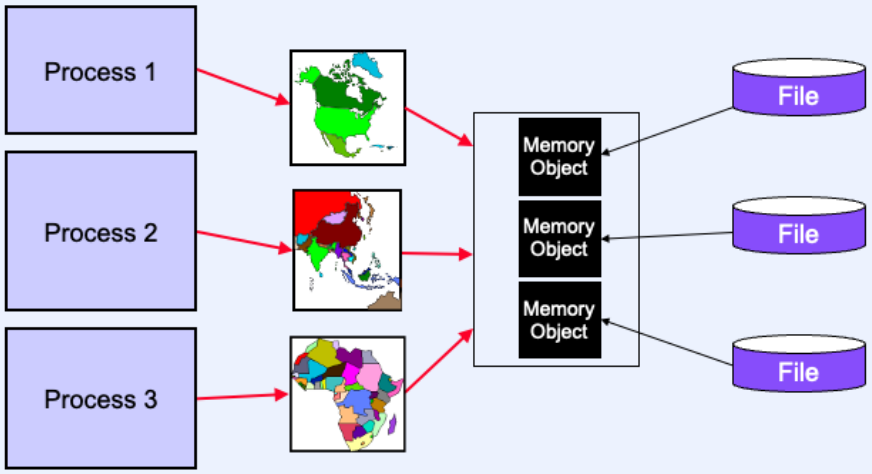
- It's on a disk!

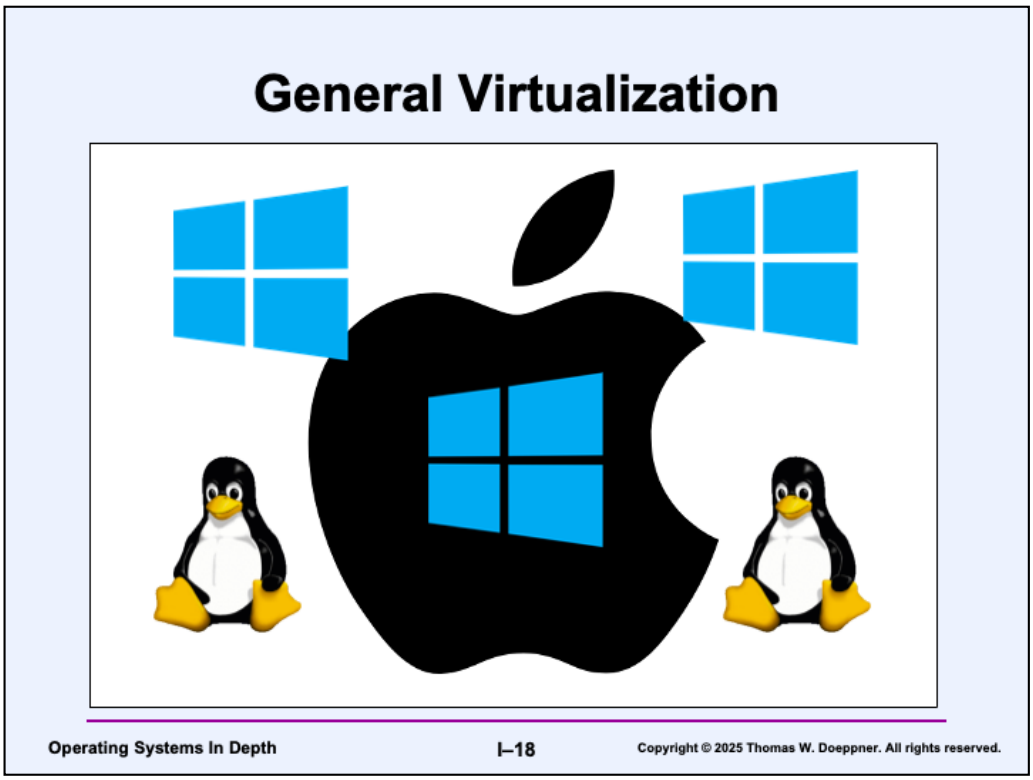


Virtual Memory

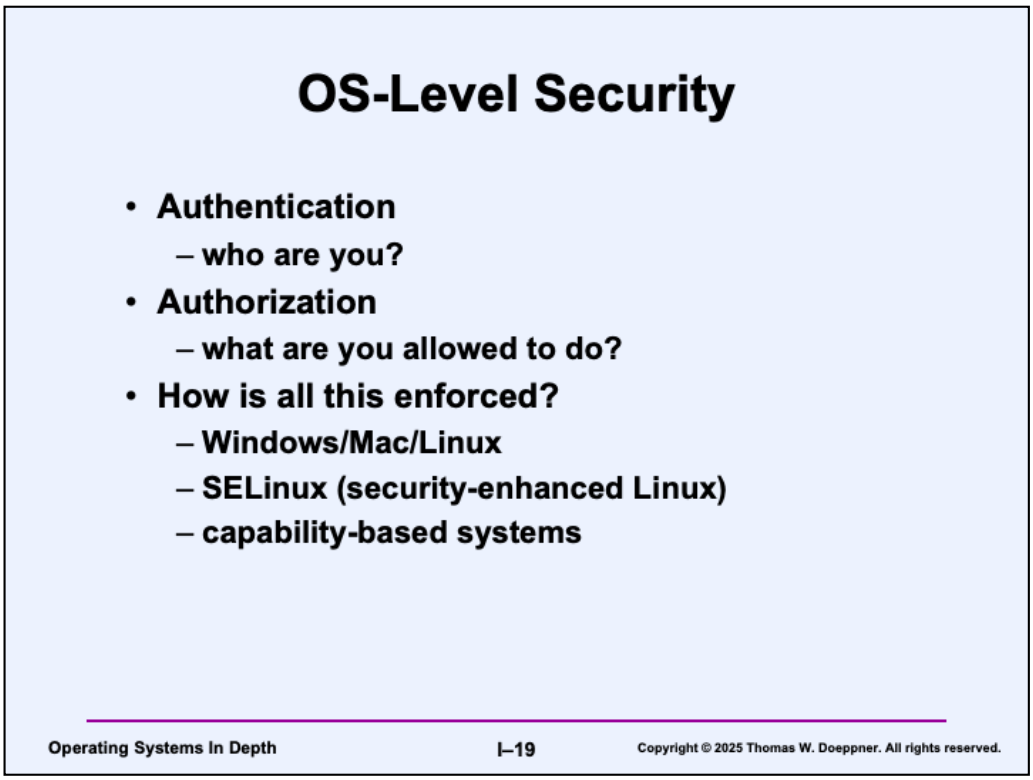


Virtual Memory + VFS

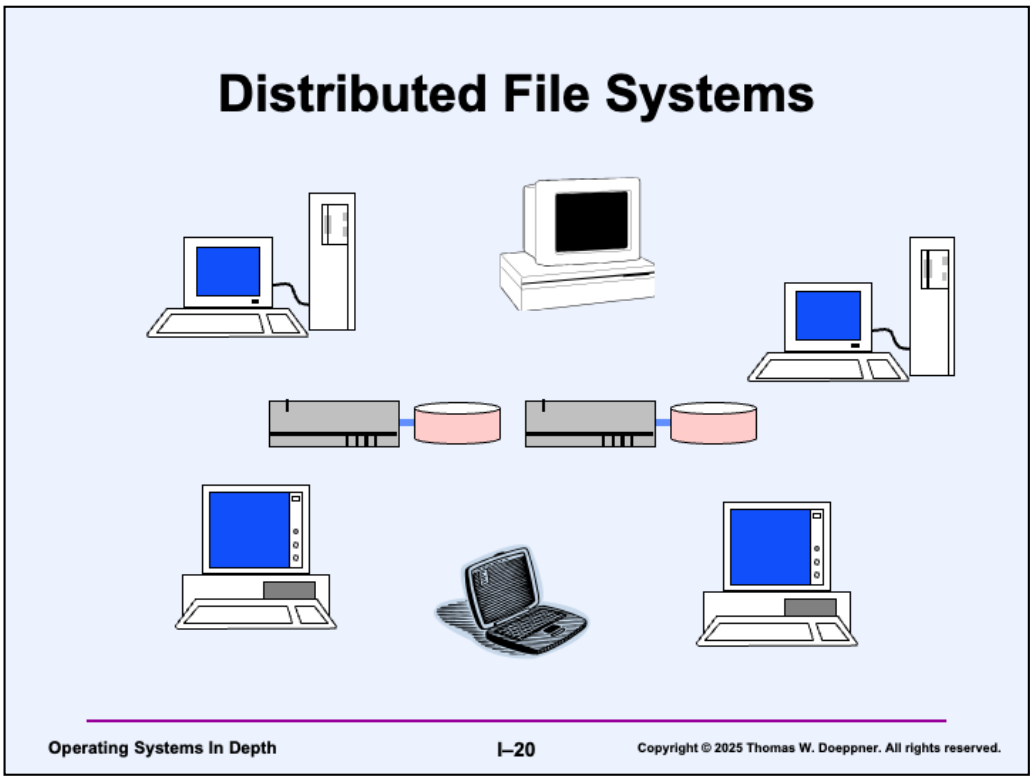




General virtualization, often known as virtual machines, is a concept invented over 50 years ago that, in the last decade or so, has become increasingly important. It allows one to run multiple "virtual computers" on one real computer, each behaving as if it were a standalone computer. On these virtual computers one can run the same operating systems as one runs on real computers, thus, for example, allowing one to run virtual computers running Windows or Linux on a real computer whose operating system is nominally OSX.



Security is an ever-increasing important aspect of operating systems. We'll look at how things are implemented in existing systems, as well as other approaches that have yet to make it to the mainstream.



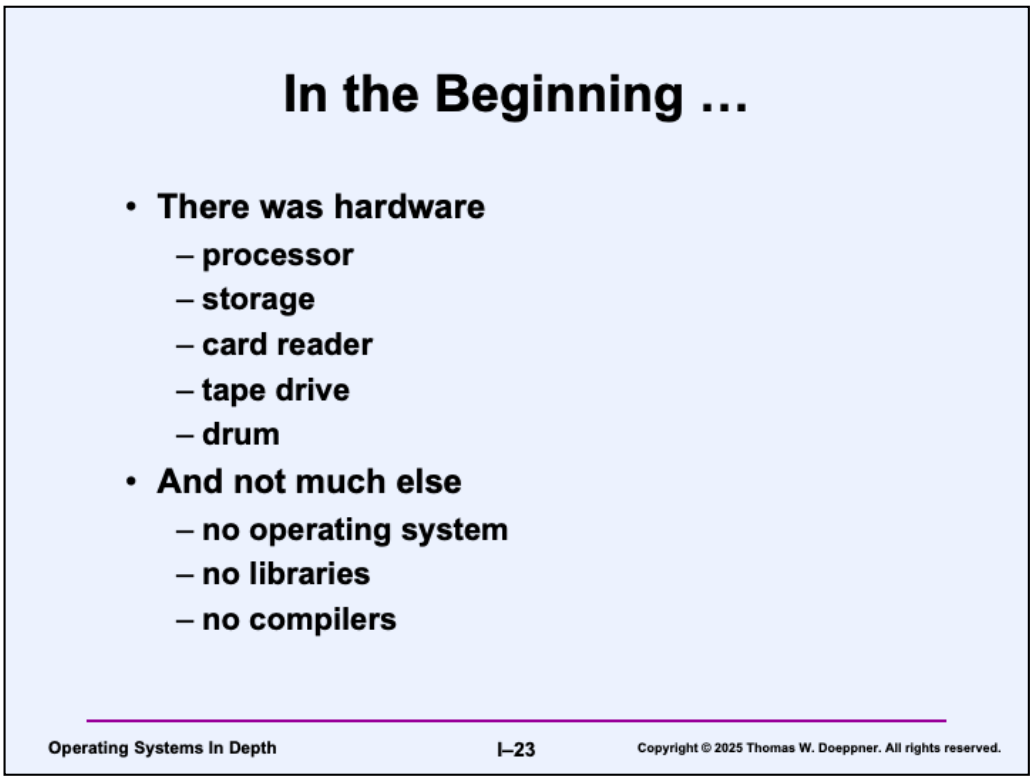
While we won't implement distributed file systems in this course, we'll definitely discuss their design and operation.

1960s OS Issues

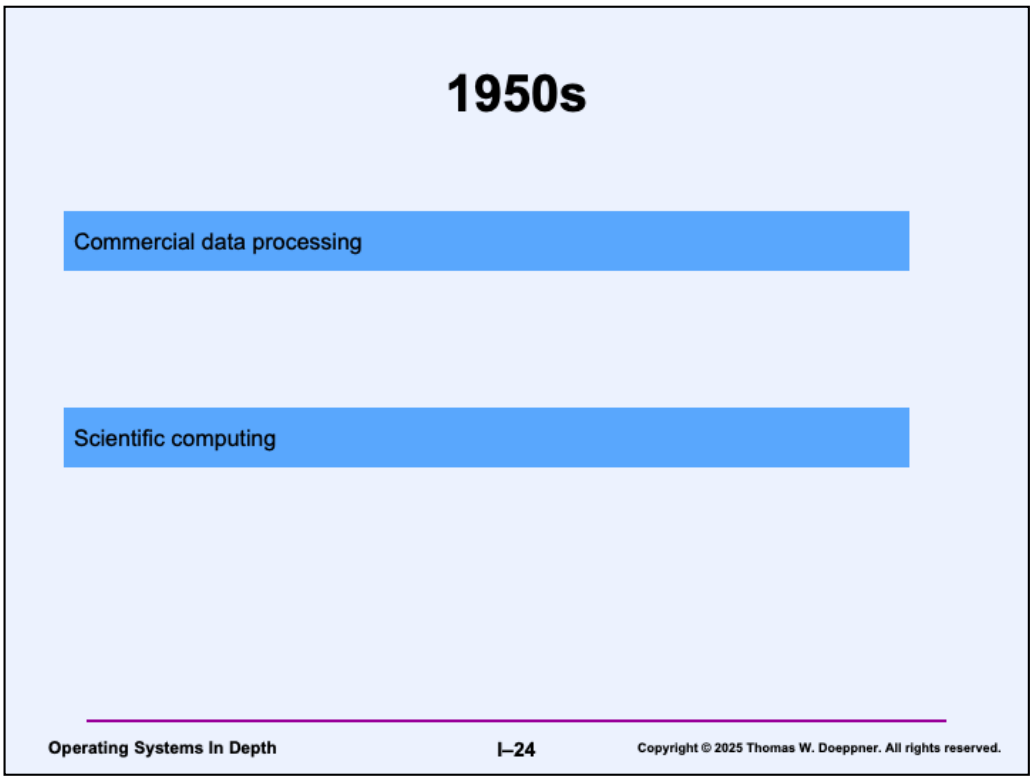
- Multiprogramming
- Time sharing
- Software complexity
- Security

2025 OS Issues

- Multiprogramming
  - not just one computer, but server farms
- Time sharing
  - voice, video, sound, etc.
- Software complexity
  - a much bigger problem than could possibly be imagined in the ’60s
- Security
  - ditto

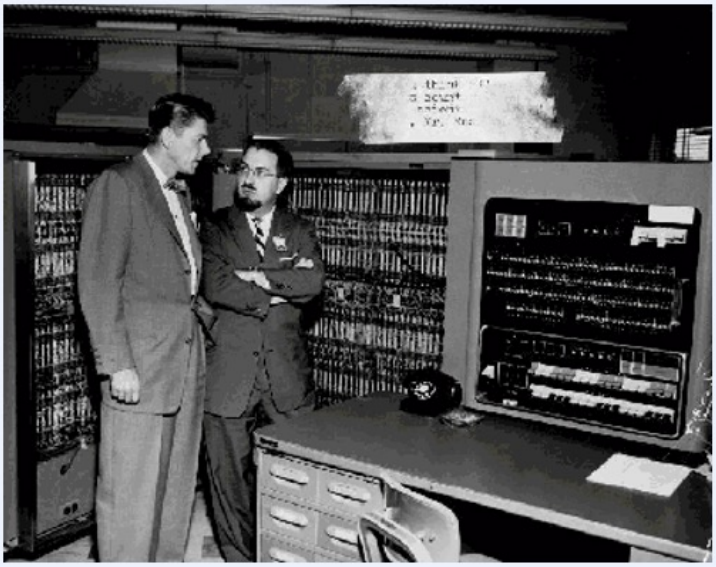


See http://www-03.ibm.com/ibm/history/exhibits/701/701_intro.html for a description and photos of the IBM 701 (a computer that was announced in April 1952). This was clearly not the first computer, but it was the first computer to have something resembling an operating system (introduced somewhat later).



Early computing was roughly split between two sorts of activities: commercial data processing and scientific computing. The former was characterized by much data processing, less computation, and "production" activities such as doing a payroll. The latter involved more computation and, generally, less data processing.

IBM 701



OS:

Initially, none

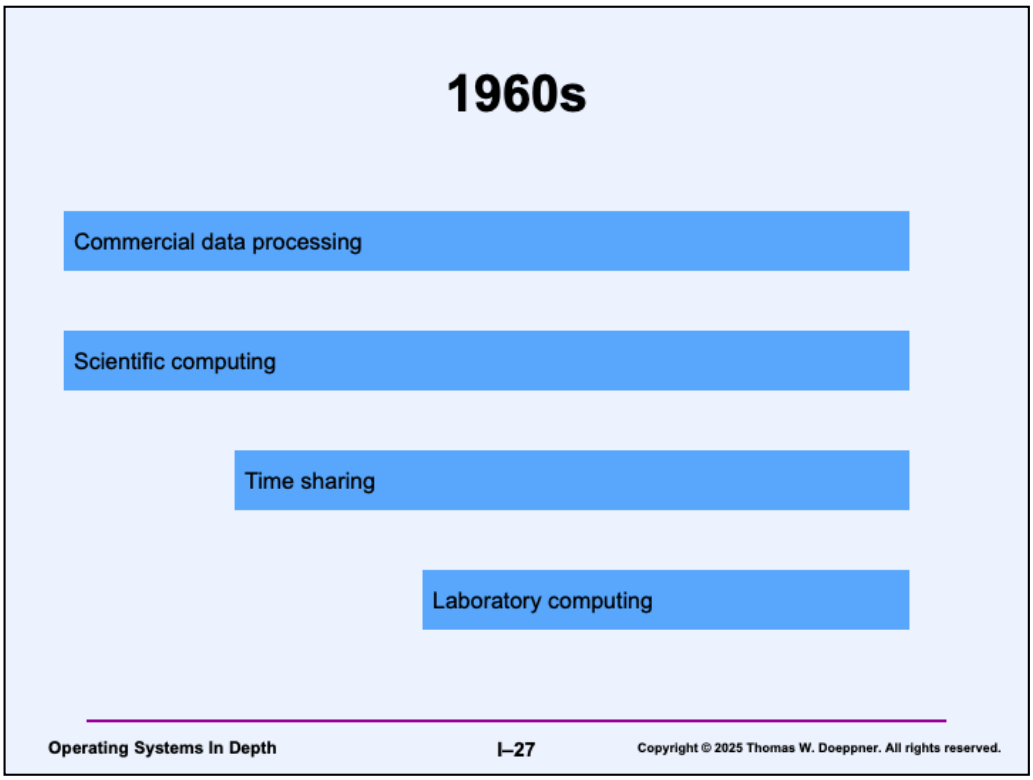
The IBM 701 is, reportedly, the first computer to have an operating system (designed by General Motors (not IBM) and called the **Input/Output System**). The man on the right is computer pioneer Herb Grosch, in whose hotel room the notion of an operating system was reportedly hashed out in 1953. He later served as president of ACM. The man on the left looks like an actor who later served as president of a large country. The picture is from http://www.fwtunesco.org/musi/museu8.html (the link no longer works), the web site of FWT UNESCO Computer Museum, Padova, Italy, and was taken in 1956.

IBM 650

OS:

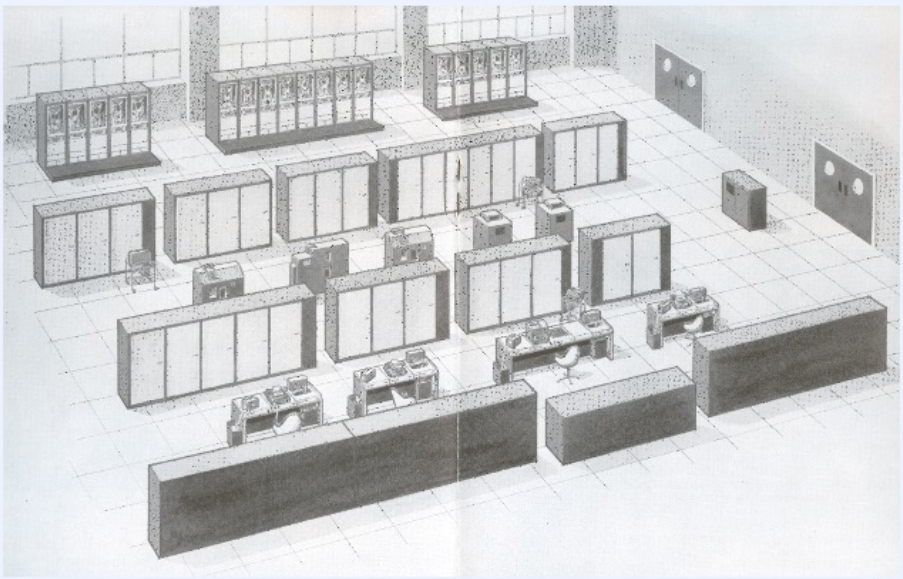
none

The photo is from http://www-03.ibm.com/ibm/history/exhibits/650/650_ph10.html and is of the IBM 650 used by the FAA for air traffic control, though it was primarily used for commercial data processing. The IBM 650 was announced in 1953 and was the most popular computer in the 1950s. It was last made in 1962 and support was terminated in 1969. It apparently had nothing resembling an operating system.



In the 1960s, commercial data processing and scientific computing continued to be important. But the decade saw the introduction of time sharing as a means for making more productive use of an expensive computer, as well as the introduction of relatively cheap minicomputers initially used in laboratories.

Atlas Computer



This drawing of a “typical Atlas computer installation” is from http://www.chilton-computing.org.uk/acl/technology/atlas/p002.htm (where there is also a diagram labeling what each of the components in the slide are). Its operating system supported multiprogramming and seems to be the first major step in OS design and implementation after the Input/Output System of the IBM 701. It’s most famous for including the first implementation of virtual memory, which they called a “single-level store.” It was a collaboration between the University of Manchester (UK) and Ferranti Ltd. It was officially commissioned in 1962, though it seems to have been in operation starting in 1961.

IBM 7094

OS:

CTSS
(among others)

This photo, of an IBM 7094, is from http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_PP7094.html. It was modified at MIT and used as the hardware platform for CTSS (compatible time-sharing system), which seems to be the first time-sharing system.

The IBM Mainframe

OS:

OS/360

The photo is of an IBM System/360 and is from http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_intro2.html. The IBM System/360 comprised a range of computers. It was most successful in the realm of commercial data processing. The earliest implementation of general virtualization was done by IBM on the 360.

Multics

This photo, of a Honeywell 6180, is from http://www.multicians.org/multics-stories.html. This machine was the successor to the GE 645 (Honeywell had purchased GE's computer business), on which Multics first ran. The 6180 incorporated many improvements and hosted the second generation of Multics.

DEC PDP-8

OS:

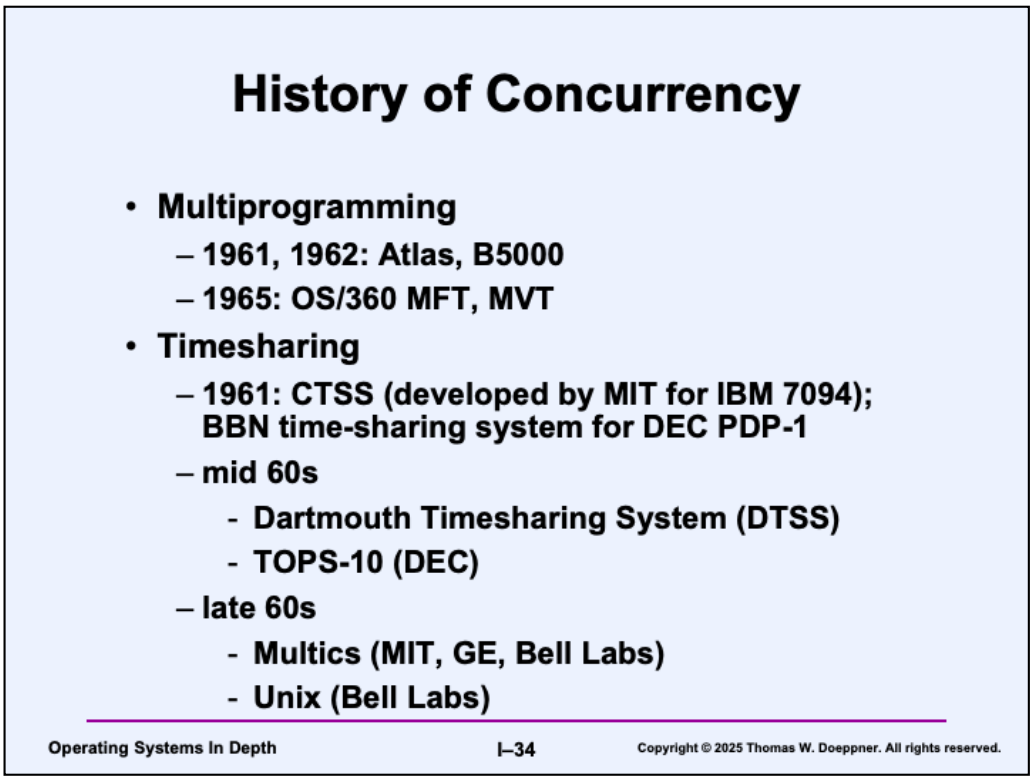
many:
ranging from primitive to interesting (a multi-user time-sharing system; a virtual-machine system)

The PDP-8, introduced in 1965 (the photo is from http://www.pdp8online.com/pdp8i/pdp8i.shtml), was the first “minicomputer” and was cheap enough to be used in small laboratories. Its manufacturer (DEC: Digital Equipment Corporation) as well as other companies produced numerous kinds of such minicomputers and an even greater number of operating systems for them. This OS development continued well into the 1970s.

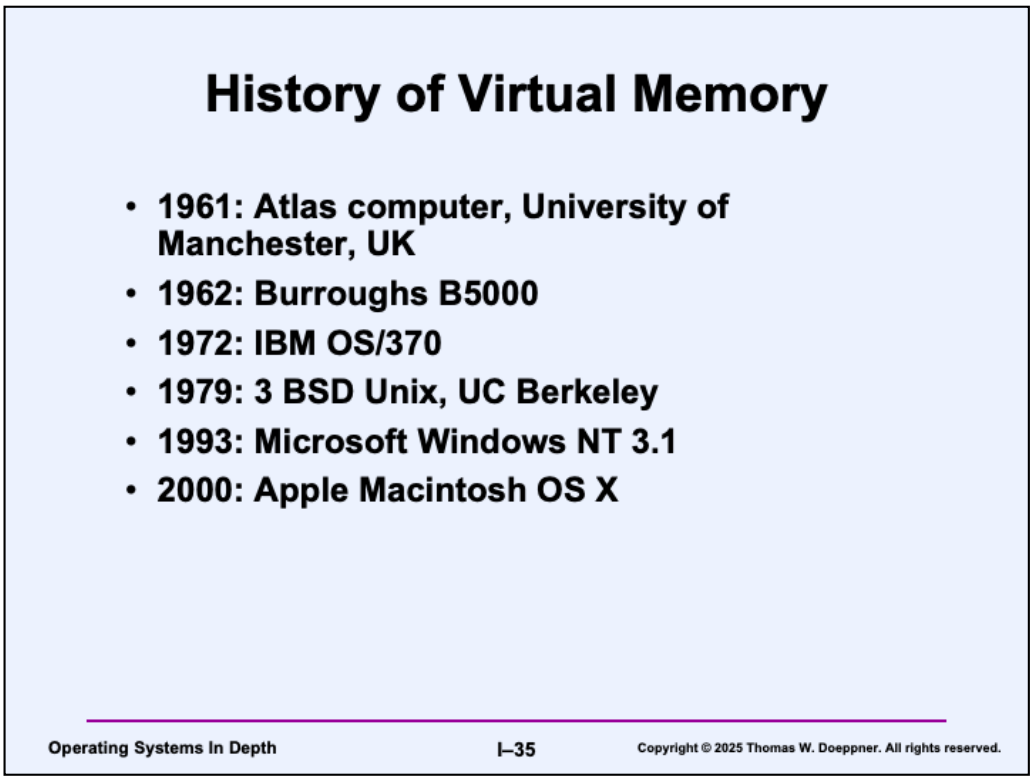
A discussion of the operating systems written for the PDP-8 can be found at http://www.faqs.org/faqs/dec-faq/pdp8/section-10.html.

Unix

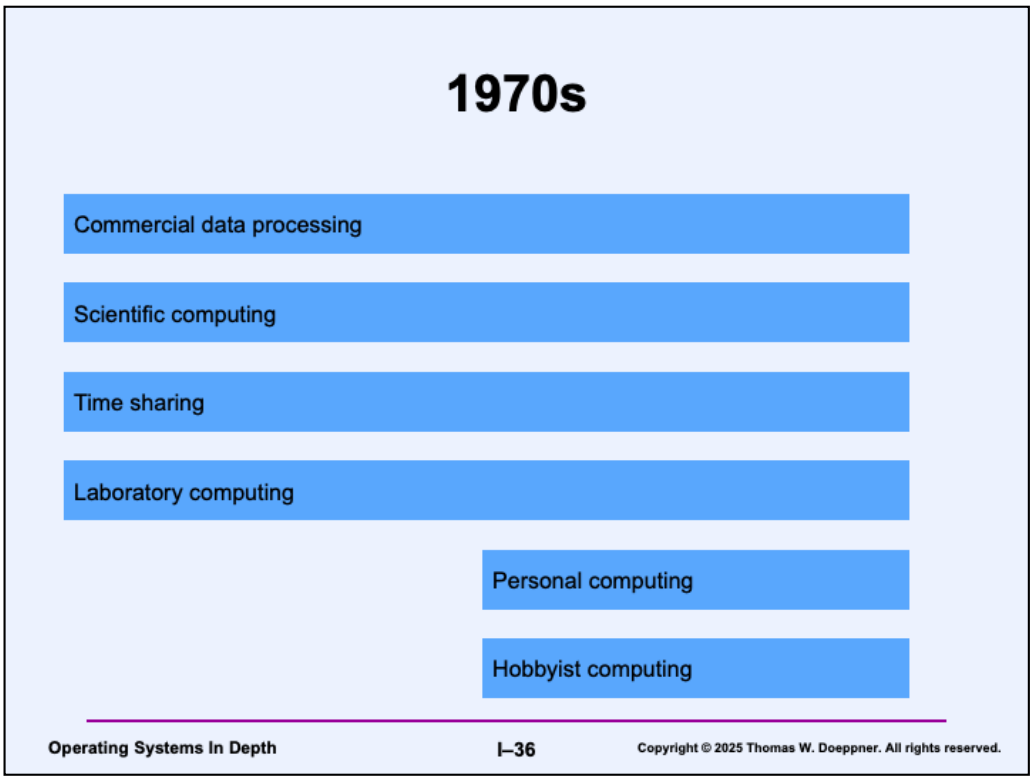
This photo is of Dennis Ritchie and Ken Thompson, the original developers of Unix, in front of a DEC PDP-11. It's from http://histoire.info.free.fr/images/pdp11-unix.jpeg, though the photo appears to be owned by Lucent Technologies. It was probably taken in the mid 1970s. Unix was originally implemented on a DEC PDP-7, but was soon ported to the more-capable PDP-11.



Multiprogramming refers to the notion of having multiple programs active at the same time so that when the current program can no longer run (for example because it's blocked waiting for I/O), another is available to run. *Timesharing* is an extension of multiprogramming in which the execution of the active programs is time-sliced: each program runs for a short period of time, then another is run. A good web site with useful information on Multics is http://www.multicians.org/. A brief description of CTSS can be found at http://www.multicians.org/thvv/7094.html.



The slide lists milestones in the history of virtual memory, from its first instance on Manchester's Atlas computer in 1961 (when a working prototype was completed), to Apple's announcement of VM support for the Macintosh.



The 1970s saw the continued importance of the sorts computing important in the ’60s. But, what would become the most significant form of computing in later decades was introduced as personal computing and hobbyist computing.

IBM's Dominance Continues

OS:

OS/370

The photo is of an IBM System/370 Model 168, from http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_2423PH3168.html. This was a high-end model and one of the first "mainstream mainframes" to support virtual memory.

Scientific Computing

OS:

COS: single job at a time

The photo is of a Cray-1 supercomputer in the Computer History Museum. The photo is by Ed Toton who, according to http://en.wikipedia.org/wiki/File:Cray-1.jpg, has released it to the public domain. The first Cray-1 was installed at Los Alamos National Laboratories in 1976. Cray's philosophy was, essentially, no compromises allowed. Virtual memory was eschewed in favor of having enough real memory to make it unnecessary. Similarly a memory cache wasn't used — all memory was as fast as most contemporary computer's caches were. The OS, known as COS (Cray Operating System) was strictly batch-oriented and designed to run one job at a time. Jobs (including associated data) were assembled by attached mainframe computers. In essence, COS did not do a whole lot more than the Input/Output System of the IBM 701.

In the 1980s, Cray adapted Unix for their machines, calling their version Unicos.

Xerox Alto

OS:

single-user, single-computation

This is a photo of a Xerox Alto from http://commons.wikimedia.org/wiki/File:Xerox_Alto_full.jpg. Produced in 1973 at Xerox's Palo Alto Research Center (PARC), it's generally regarded as the first computer to demonstrate the utility of a window manager and mouse. Though it was never a product, it introduced the notion of a serious personal computer.

Though its historical importance is primarily in its pioneering use of bit-mapped graphics to implement a window-managed display, it had an interesting OS as well. It was strictly a single-user system and provided no protection whatsoever. However, it had a layered approach to OS design and implementation. Through its notion of "juntas," one could remove layers of the OS, one at a time, just in case one needed the storage for a large program (and one could read them back from disk).

A great deal of Alto documentation can be found at http://www.bitsavers.org/pdf/xerox/alto/.

MITS ALTAIR 8800

OS:

none

This photo, of a MITS ALTAIR 8800, is from http://www.vintage-computer.com/altair8800.shtml. Introduced in 1975, it was an early, if not the first, "hobbyist computer." It was also the platform on which Microsoft's first product, a BASIC interpreter, ran. It had no operating system, just standalone programs (in particular, a BASIC interpreter supplied by Microsoft — their very first product and very first platform).

CP/M

- Control Program for Microcomputers
  - 1974
  - first hobbyist OS
  - supported Intel 8080 and other systems
    - clear separation of architecture-dependent code
  - no multiprogramming
  - no protection

CP/M was the first OS affordable to individual hobbyists.

Apple II

OS:

initially: none
later: similar functionality as CP/M (not much)

This photo, of an Apple II, is from http://commons.wikimedia.org/wiki/File:Apple_II.jpg. It was introduced in 1977 with no OS. Later (in 1978) a simple OS was released with functionality similar to that of CP/M.

There was an Apple I, designed and built by Steve Wozniak in 1976. Only 200 were made; many were traded in for Apple II's (a trade which Apple did for free, so as to free up Steve Wozniak for other work). It ran a Basic interpreter.

Microsoft Enters the OS Business: Late 1970s

- It's called …
  - Xenix
    - a version of Unix
  - predominant version of Unix in the 1980s
  - used by MS internally into the 1990s

VAX-11/780

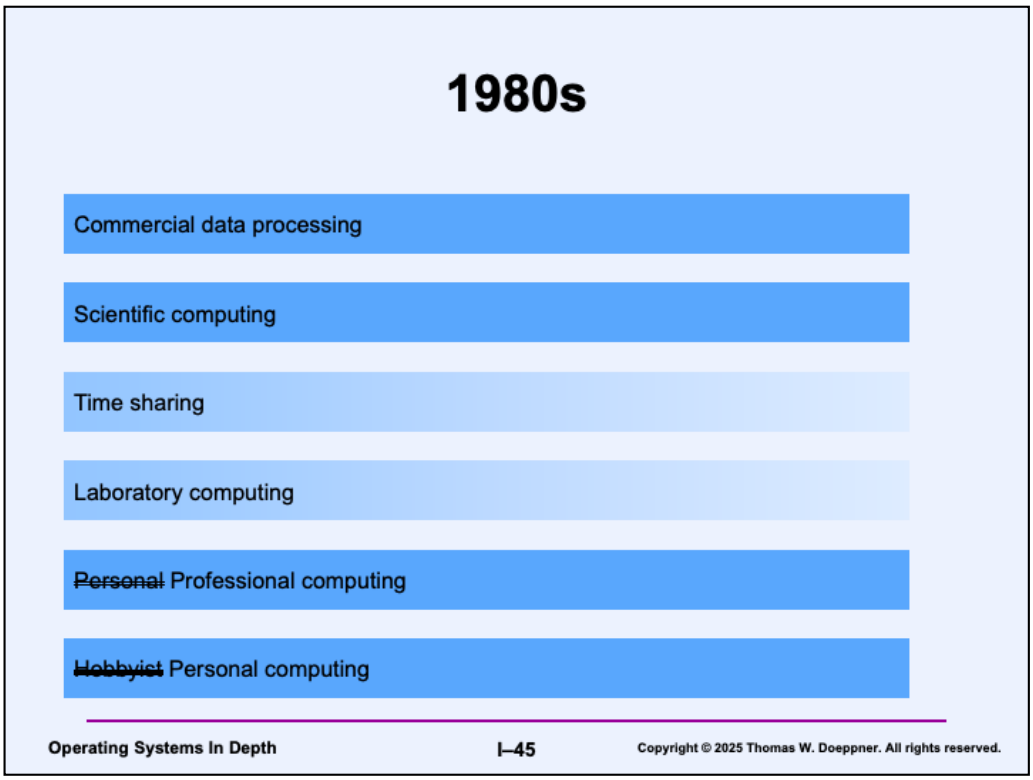
OS:

VMS
Unix

Both:
- **time sharing**
- **virtual memory**
- **access protection**
- **concurrency**

The VAX-11, of which the 780 was the first model, was introduced in 1978 and was noted for two operating systems. The first, VMS, was the product of its manufacturer, DEC. It is still supported by HP, the company that purchased Compaq, which was the company that purchased DEC. VMS was very much the predecessor of modern Windows.

The second operating system is Unix, most notably BSD Unix. Seventh-Edition Unix was ported to the VAX-11 by Bell Labs and called Unix 32V. It did not support virtual memory. Researchers at the University of California at Berkeley used it as the basis of 3BSD Unix, which did support virtual memory. Later came 4.0BSD, 4.1BSD, up through 4.4BSD. Along the way DEC produced its own version of Unix for the VAX-11, based on 4.2 BSD, called Ultrix. It was the implementation of the TCP/IP network protocols on 4.2BSD that did much to grow the Internet, making it accessible to a large number of academic computer science departments and industrial research organizations.



The introduction of the IBM PC co-opted the term "personal computing," which previously had been applied to work done at Xerox PARC. High-end "computer workstations" were introduced for professional computing, running more-or-less state-of-the-art operating systems. More affordable computers, such as IBM PCs (and clones) and Apple Macintoshes, had much less capable operating systems.

The advent of the PC pretty much eliminated the mini-computer market and separate computers (and operating systems) for laboratory computing.

Two OSes Take Off

Unix

MS-DOS

The 1980s saw the rise of two operating systems: Unix via BSD Unix on VAXes and Xenix on higher-end PCs, and MS-DOS on PCs.

IBM PC

OS:

PC-DOS
(aka MS-DOS)
(remarkably like CP/M)

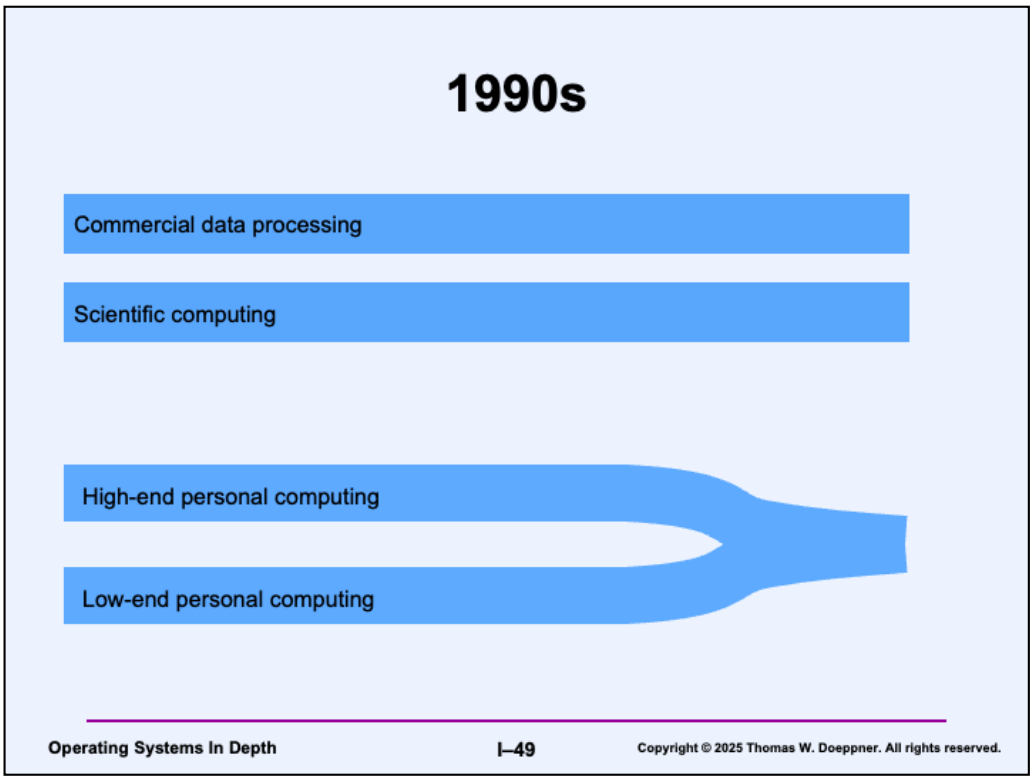
This photo of an early IBM PC is from http://www-03.ibm.com/ibm/history/history/history_intro.html.

The Computer Workstation

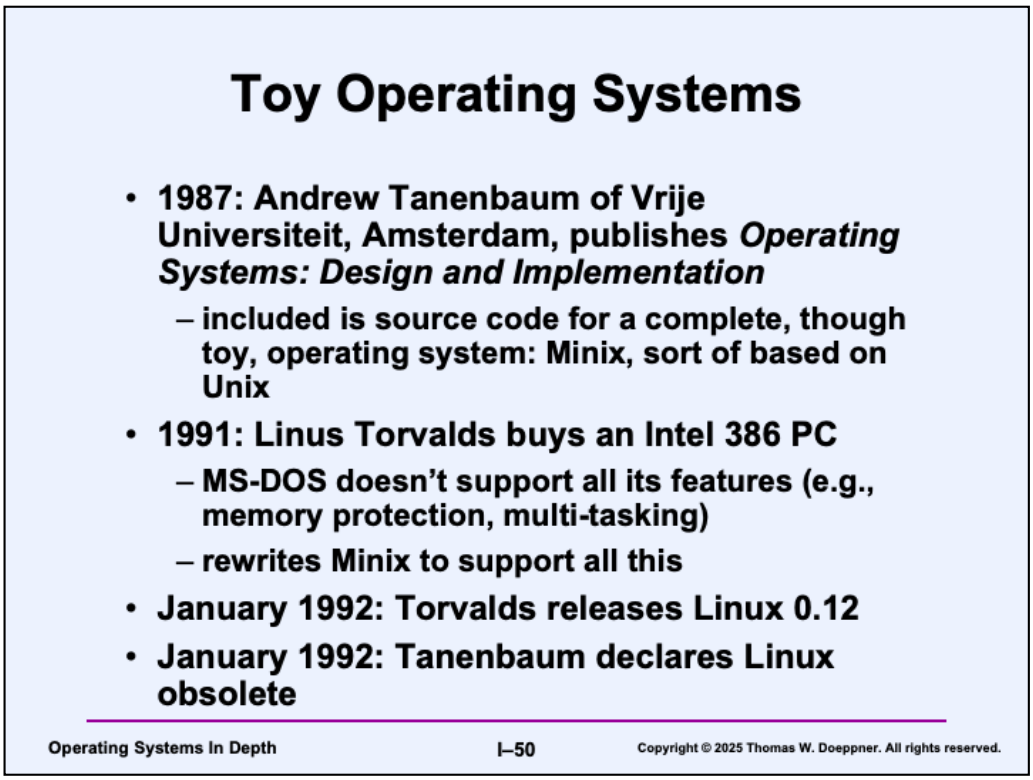
OS:

Aegis
supported:
- **virtual memory**
- **distributed file system**
- **access protection**
- **concurrency**

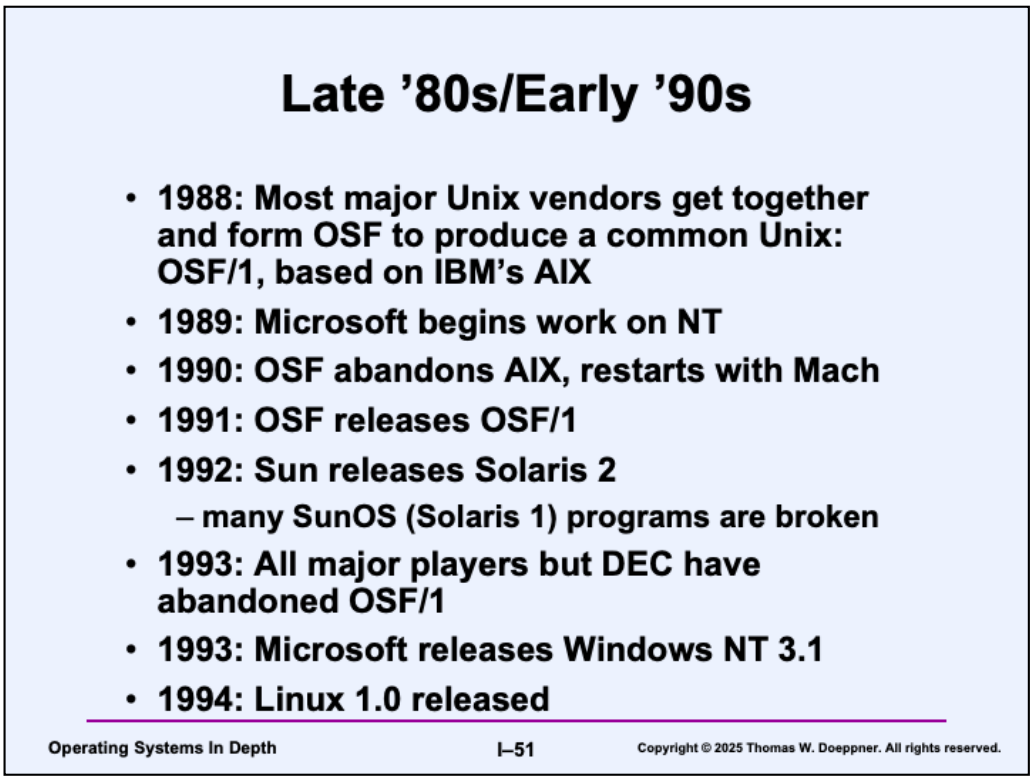
An Apollo workstation, introduced in 1982. (Note the “C” on its front — this was its serial number (in hexadecimal)). It had a fairly sophisticated OS for its day.



The 1990s saw the convergence of the low- and high-end personal computing: hardware powerful enough to run the operating systems of high-end personal computing became affordable at the low-end.



Linux was originally named "Freax" by Torvalds, but a friend of his came up with "Linux".

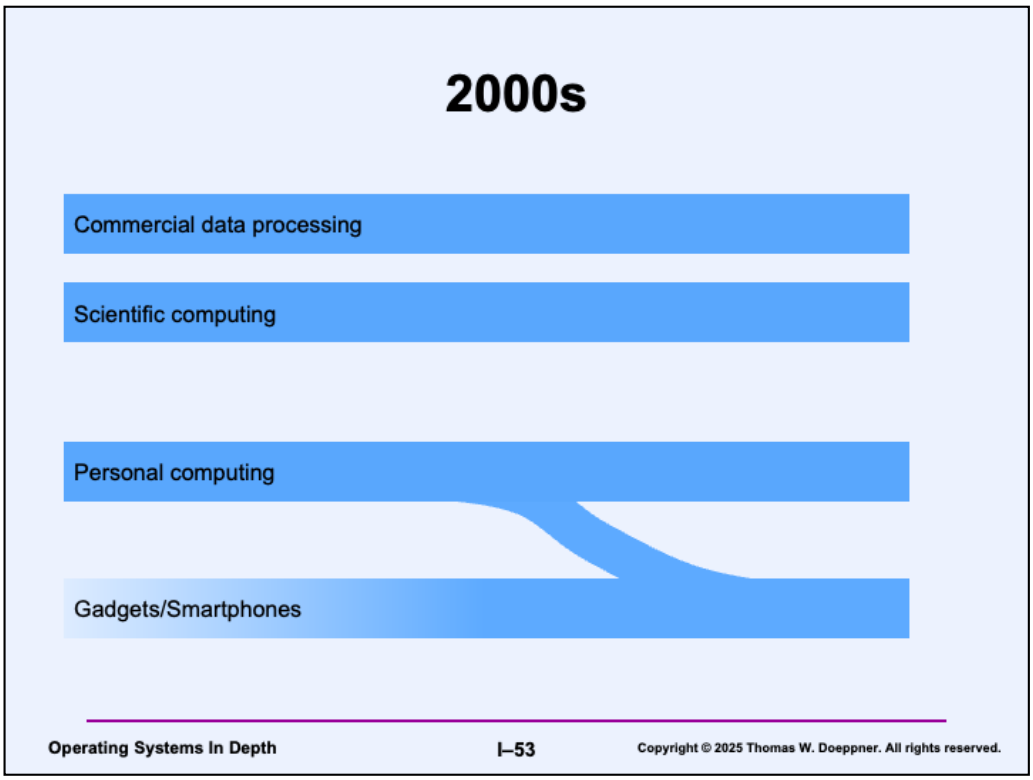


Note that the notion of a completely free (and useful) operating system didn’t really exist in the early 1990s. Unix was licensed by AT&T, who charged a hefty fee for its commercial use.

Windows NT 3.1 was the first release of Windows NT. It was numbered 3.1 because the version of regular (non-NT) Windows was 3.1 at the time, and Microsoft didn’t want to confuse people by having multiple current version numbers.

Late ’90s

- 1996: DEC renames its OSF/1 “Digital Unix”
- 1996: Microsoft releases Windows NT 4
- 1996: Linux 2.0 released
- 1998: DEC is purchased by Compaq; “Digital Unix” is renamed “Tru64 Unix”
- 1999: Sun’s follow-on to Solaris 2.6 is called Solaris 7



The 2000s have brought on the “gadget” as an important computing device, one deserving of an OS.

The ’00s Part 1

- 2000: Microsoft releases Windows 2000 and Windows Me
- 2000: Linux 2.2 is released
- 2000: IBM “commits” to Linux (on servers)
- 2001: Apple releases Mac OS X, based on Unix and Mach (sort of like OSF/1)
- 2001: Linux 2.4 is released
- 2001: Microsoft releases Windows XP
- 2002: Compaq is purchased by HP
- 2003: SCO claims their code is in Linux, sues IBM; IBM countersues
  - August 10, 2007: judge rules that SCO is not the rightful owner of the Unix copyright, Novell is
  - Novell says there is no Unix in Linux
  - September 2007: SCO files for Chapter 11 bankruptcy protection

The ’00s Part 2

- 2004: Linux 2.6 is released
- 2005: IBM sells PC business to Lenovo
- July 2005: Microsoft announces Windows Vista
- January 2007: Microsoft releases Windows Vista
- Later in 2007: Microsoft starts hinting at Windows 7
- June 2007: Apple releases iOS for iPhone
- April 2009: Oracle announces purchase of Sun Microsystems
- July 2009: Google announces Chrome OS
- October 2009: Microsoft releases Windows 7

The ’10s Part 1

- January 2011: Microsoft announces Windows 8
- June 2011: first products shipped running Chrome OS
- April 2011: Linux 3.0 released
- July 2011: Apple drops “Mac” prefix for its OS: it’s now “OS X”
  - one million copies of OS X Lion sold on first day of release
- October 2011: deaths of both Dennis Ritchie and Steve Jobs
  - “Dennis Ritchie: The Shoulders Steve Jobs Stood On”
    - Wired Magazine, 10/13/2011
- October 2012: Microsoft releases Windows 8

The ’10s Part 2

- September 2014: Microsoft Announces Windows 10
  - Windows 9 skipped because it might be confused with Windows 95
- October 2014: Apple releases OS X 10.10 Yosemite
  - doesn’t work with Brown University wi-fi
  - November 2014 release of 10.10.1 doesn’t fix it ...
- April 2015: Linux 4.0 released
- July 2015: Microsoft Releases Windows 10
  - Brown’s CIS says “don’t switch yet”
  - August 2015: Brown’s card-access system switched to Windows 10
  - August 2015: Brown’s card access fails university-wide
- September 2015: Apple releases OS X 10.11 El Capitan
  - Brown’s CIS says “don’t switch yet”

The ’10s Part 3

- September 2016: Apple releases OS X 10.12 Sierra
  - Brown CIS says don’t switch
  - Brown CS tstaff says printing is broken
- December 2016: Apple releases OS X 10.12.2
  - twd switches
  - his D-Link router no longer supported printing
  - his Logitech mouse’s scroll wheel scrolled window in random directions
- January 2018: Meltdown and Spectre vulnerabilities disclosed
  - end of computing (if not civilization) as we know it?

The ’10s Part 4

- September 2018: Apple releases OS X 10.14 Mojave
  - no one switches
- October 2018: Apple releases OS X 10.14.1
  - twd switches
  - PowerPoint presentations in CS33 randomly switch to random photos
- March 2019: Linux 5.0 released
- October 2019: Apple releases OS X 10.15 Catalina

The ’20s Part 1

- November 2020: Apple releases OS X 11.0 Big Sur
  - twd upgraded his laptop
  - CIS-supported antivirus software won’t let machine shut down
    - fixed in 11.1
- October 2021: Apple releases OS X 12.0 Monterey
  - January 2022: OIT (formerly CIS) finally says it’s ok to use it
  - twd bought a new laptop that came with Monterey (12.0.1) installed
  - twd’s 7-year-old iMac was too old to run it
- October 2022: Apple releases OS X 13.0 Ventura
  - CSCI 330/1330 students who installed it regretted it

See https://ithelp.brown.edu/kb/articles/ventura-macos-13-what-works for a list of what works on Ventura.

The ’20s Part 2

- September 2023: Apple announces OS X 14.0 Sonoma
  - OIT says many things don’t work
  - twd waits till 14.2.1 before switching
    - Logitech mouse’s tracking speed can’t be adjusted
- September 2024: Apple announces macOS 15.0 Sequoia
  - macOS reintroduced to differentiate it from iOS
  - twd gets new desktop with 15.0
    - no major problems!
    - now running 15.1.1

Friday

- Implementing threads
  - now is a good time to review POSIX threads