

System Architecture

History of Operating Systems

Some slides from A. D. Joseph, University of Berkeley

See also:

www.osdata.com/kind/history.htm

www.armory.com/~spectre/tech.html

courses.cs.vt.edu/~cs1104/VirtualMachines/OS.1.html

en.wikipedia.org/wiki/History_of_operating_systems



Moore's Law Drives OS Change

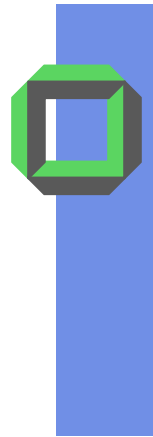
	1981	2006	Factor
CPU MHz, Cycles/inst	10 3—10	3200x4 0.25—0.5	1,280 6—40
DRAM capacity	128KB	4GB	32,768
Disk capacity	10MB	1TB	100,000
Net bandwidth	9600 b/s	1 Gb/s	110,000
# addr bits	16	32	2
#users/machine	10	≤ 1	≤ 0.1
Price	\$25,000	\$4,000	0.2

Typical academic computer 1981 vs 2006

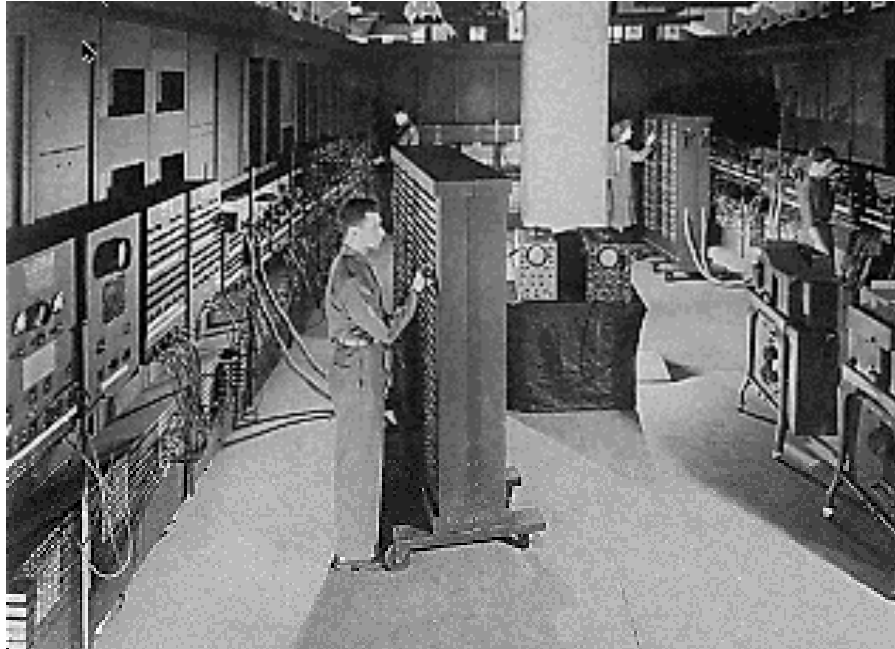


Moore's Law Effects

- Nothing like this in any other area of business
- Transportation in over 200 years:
 - Only 2 orders of magnitude from horseback @10mph to Concorde @1000mph
 - Computers do this every decade
- What does this mean for us?
 - Techniques have to vary over time to adapt to changing tradeoffs
- Let's place a lot more emphasis on principles
 - The key concepts underlying computer systems
 - Less emphasis on facts that are likely to change over the next few years...
- Let's examine the way changes in \$/MIP has radically changed how OS's work



Dawn of Time ENIAC: (1945-55)

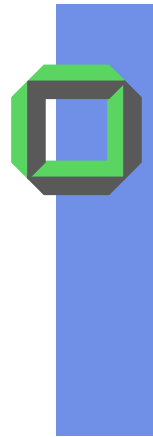


- “The machine designed by Eckert and Mauchly was a monstrosity. When it was finished, the ENIAC filled an entire room, weighed **30 tons**, and consumed **200 kilowatts** of power.”
- <http://ei.cs.vt.edu/~history/ENIAC.Richey.HTML>

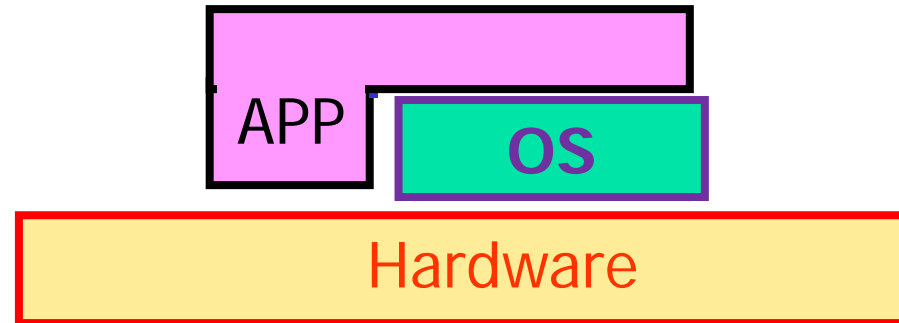


History Phase 1: 1948-70

Expensive Hardware
Cheap Humans



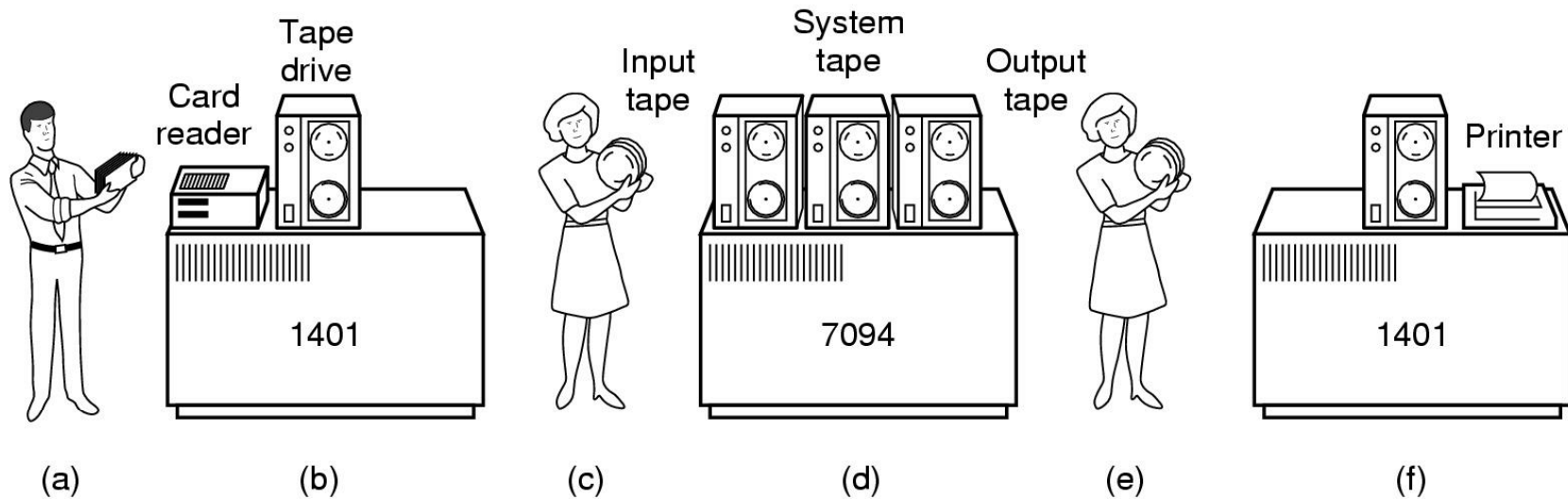
History OS: Evolution Step 0



- Simple OS: One program, one user, one machine:
 - examples: early computers, early PCs,
 - embedded controllers such as Nintendo, cars, elevators
 - OS just a library of standard services, e.g. standard device drivers, interrupt handlers, I/O
- Non-problems: **No malicious people. No bad programs**
⇒ A minimum of complex interactions
- Problem: poor utilization, expensive



History of Systems



- Early batch system
 - bring cards to 1401
 - read cards to tape
 - put tape on 7094 which does computing
 - put tape on 1401 which prints output

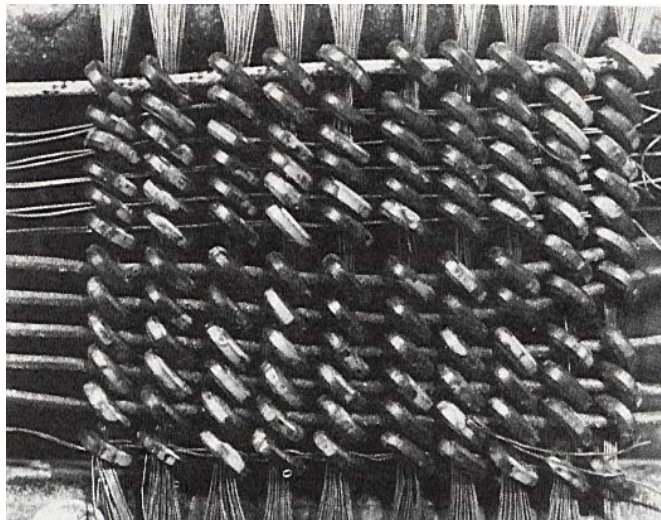


History Phase 1

- When computers cost millions of \$'s, optimize for more efficient use of the hardware
 - Lack of interaction between user and computer
- User at console: one user at a time
- Batch monitor: load program, run, print
- Optimize to better use hardware
 - When user thinking at console, computer idle⇒ very bad
 - Feed computer batches and make users wait
- **No protection**: what if batch program was buggy?



Core Memories (1950s & 60s)



The first magnetic core memory, from the IBM 405 Alphabetical Accounting Machine.

- Core Memory stored data as magnetization in iron rings
 - Iron “cores” woven into a 2-dimensional mesh of wires
 - Origin of the term “Dump Core”
 - Rumor that IBM consulted Life Saver company
- <http://www.columbia.edu/acis/history/core.html>



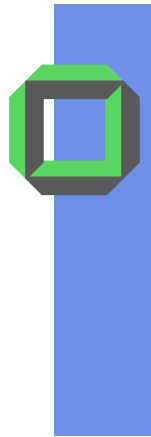
Late 60s - Early 70s

- **Data channels, Interrupts:** overlap I/O and compute
 - DMA – Direct Memory Access for I/O devices
 - I/O can be completed asynchronously
- **Multiprogramming:** $n > 1$ programs run simultaneously
 - Small jobs not delayed by large jobs
 - More overlap between I/O and CPU
 - Need memory protection between programs and/or OS



Late 60s - Early 70s

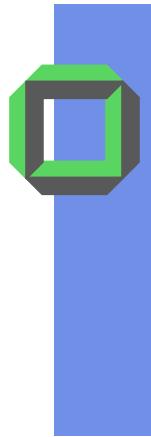
- Complexity starts to get out of hand:
 - Multics: announced in 1963, ran in 1969
 - **1777** people “contributed to MIT’s Multics” (30-40 core dev)
 - Turing award lecture from Fernando Corbató (key researcher): “On building systems that will fail”
 - OS 360: released with **1000** known bugs
 - “Anomalous Program Activity Report”
- OS finally becomes an important science:
 - *How to deal with complexity?*
 - Result: UNIX based on Multics, but vastly simplified



The Multics System (~ 1976)



- The 6180 at MIT IPC, skin doors open, circa 1976:
 - “We usually ran the machine with doors open so the operators could see the AQ register display, which gave you an idea of the machine load, and for convenient access to the EXECUTE button, which the operator would push to enter BOS if the machine crashed.”
- <http://www.multicians.org/multics-stories.html>



Ritchie & Thompson at PDP 11



Unix needed 16 KB*

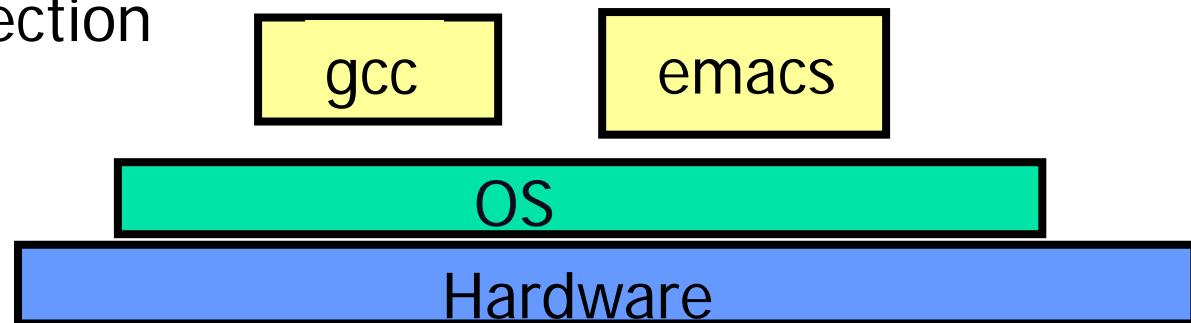
users could only get 8 KB
for their application

*at that time a mini-OS



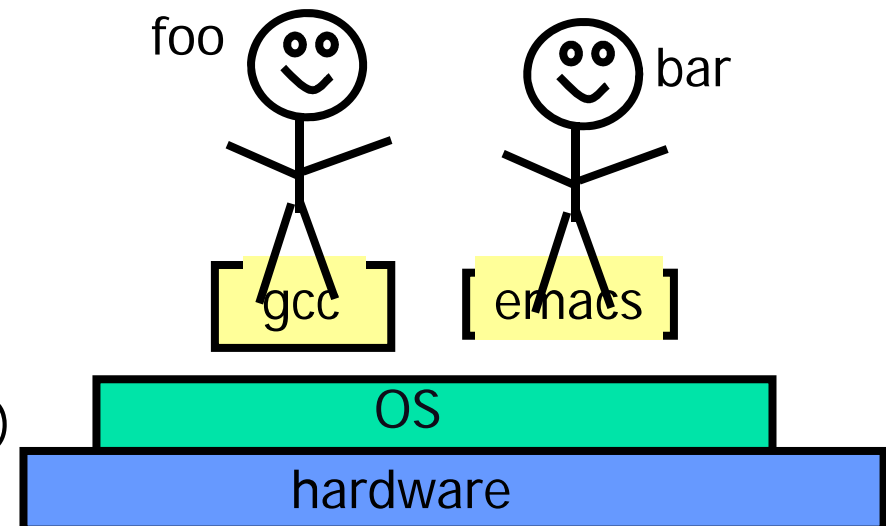
History OS: Evolution Step 1

- Simple OS is inefficient:
 - a waiting process blocks everything else on the machine
- (Seemingly) Simple hack:
 - run more than one process at once
 - when one process blocks, switch to another
- A couple of problems: *what if a program*
 - *does infinite loops or*
 - *starts randomly scribbling on memory?*
- OS adds protection
 - Interposition
 - Preemption
 - Privilege



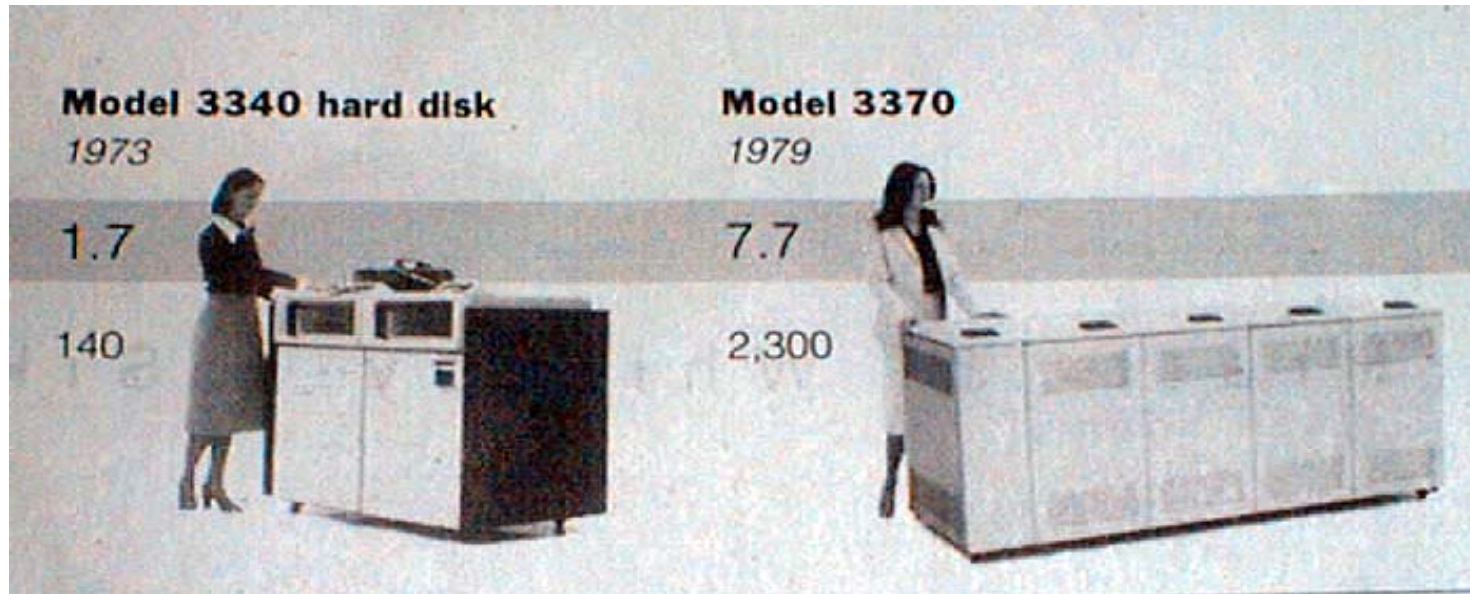
History OS: Evolution Step 2

- Simple OS is too expensive:
 - one user = one computer \Rightarrow
- (Seemingly) simple hack:
 - Allow more than one user at once
 - *Does machine now run n times slower?* Usually not
 - Key observation: users are active in bursts
 - If idle, give resources to others
- Problems: *what if*
 - *users are greedy*
 - *evil*
 - *or just too numerous?*
- OS adds protection
 - (notice: as we try to utilize resources, complexity grows)





Early Disk History



1973:
1.7 Mbit/sq. in
140 MBytes

1979:
7.7 Mbit/sq. in
2300 MBytes

source: New York Times, 2/23/98, page C3,
"Makers of disk drives crowd even more data into even smaller spaces"



History Phase 2: 70 - 85

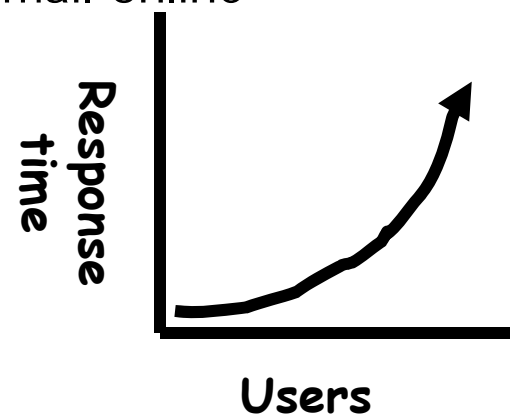
Cheaper HW

More Expensive Humans



History Phase 2

- Computers available for tens of thousands of dollars instead of millions
- OS Technology maturing/stabilizing
- Interactive timesharing:
 - Use cheap terminals (~\$1000) to let multiple users interact with the system at the same time
 - Sacrifice CPU time to get better response time
 - Users do debugging, editing, and email online
- Problem: Thrashing
 - Performance very non-linear response with load
 - Thrashing caused by many factors including
 - Swapping
 - Inefficient queuing





History Phase 3: 81 - 89

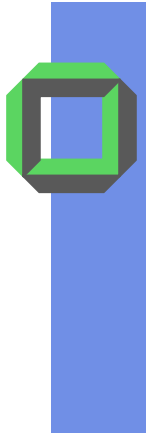
Very Cheap HW

Very Expensive Humans



History Phase 3 (1981—)

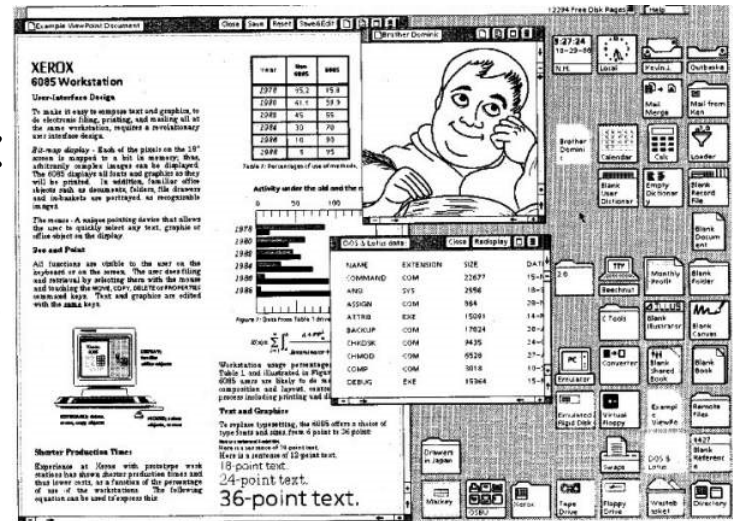
- Computer costs \$1K, Programmer costs \$100K/year
 - If you can make someone 1% more efficient by giving them a computer, it's worth it
 - Use computers to make people more efficient
- Personal computing:
 - Computers cheap, so give everyone a PC
- Limited Hardware Resources Initially:
 - OS becomes a subroutine library
 - One application at a time (MSDOS, CP/M, ...)
- Eventually PCs become powerful:
 - OS regains all the complexity of a "big" OS
 - multiprogramming, memory protection, etc (NT, OS/2)
- Question: *As HW gets cheaper, does the need for OS research go away?*



Graphical User Interfaces

- CS160 ⇒ All about GUIs
- Xerox Star: 1981
 - Originally a research project (Alto)
 - First "mice", "windows"
- Apple Lisa/Machintosh: 1984
 - "Look and Feel" suit 1988
- Microsoft Windows:
 - Win 1.0 (1985) } Single Level
 - Win 3.1 (1990) } Single Level
 - Win 95 (1995) } Single Level
 - Win NT (1993) } HAL/Protection
 - Win 2000 (2000) } No HAL/Full Prot
 - Win XP (2001) } No HAL/Full Prot

Xerox Star



Windows 3.1

