Overview
Introduction to Operating Systems

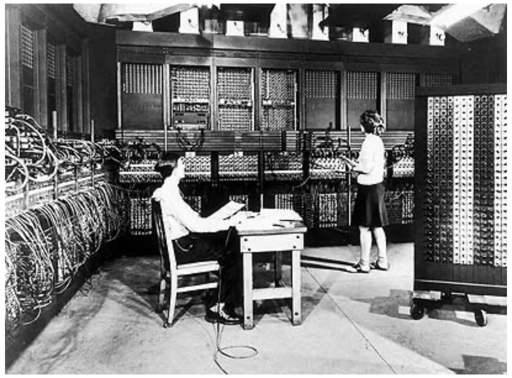
Some of the following slides are adapted from Matt Welsh, Harvard Univ.

In the Beginning…

- **There was no OS – just libraries**
  - Computer only ran one program at a time, so no need for an OS
  - Programming through wiring..

Harvard Mark I, 1944

ENIAC, 1945

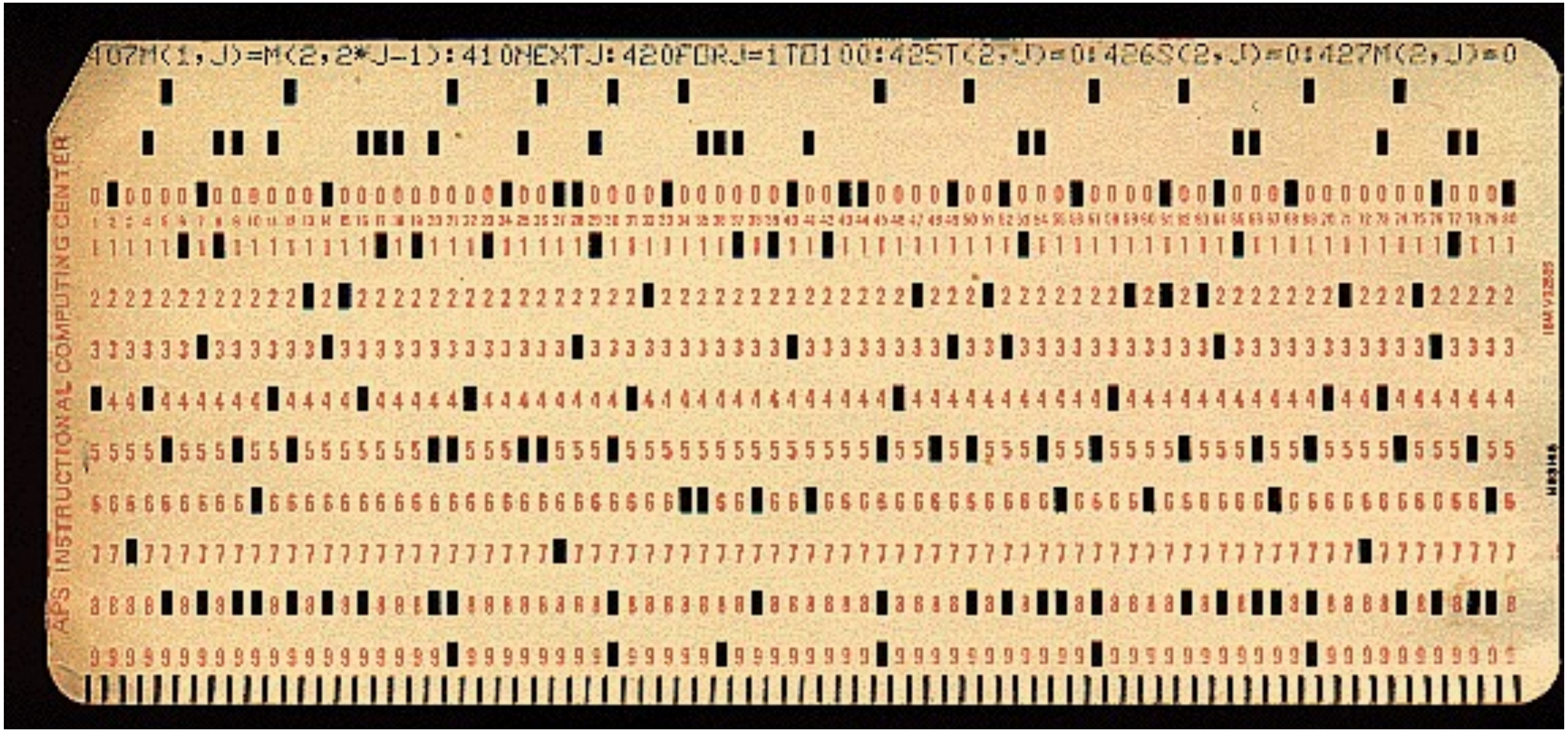


IBM 360, 1960's

In the Beginning…

- **There was no OS – just libraries**
  - Computer only ran one program at a time, so no need for an OS
- **And then there were batch systems**
  - Programs printed on stacks of punchhole cards
  - OS was resident in a portion of machine memory
  - When previous program was finished, OS loaded next program to run

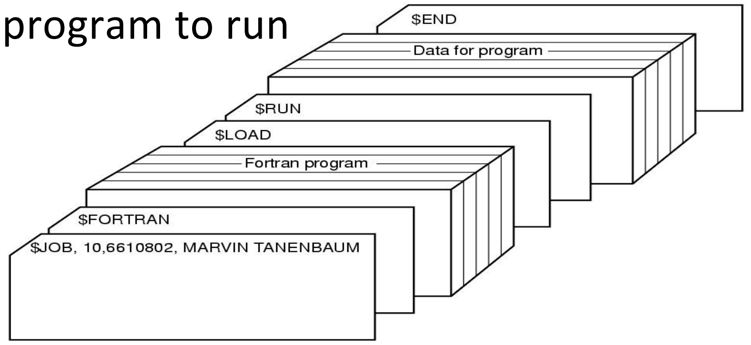
Punch Card

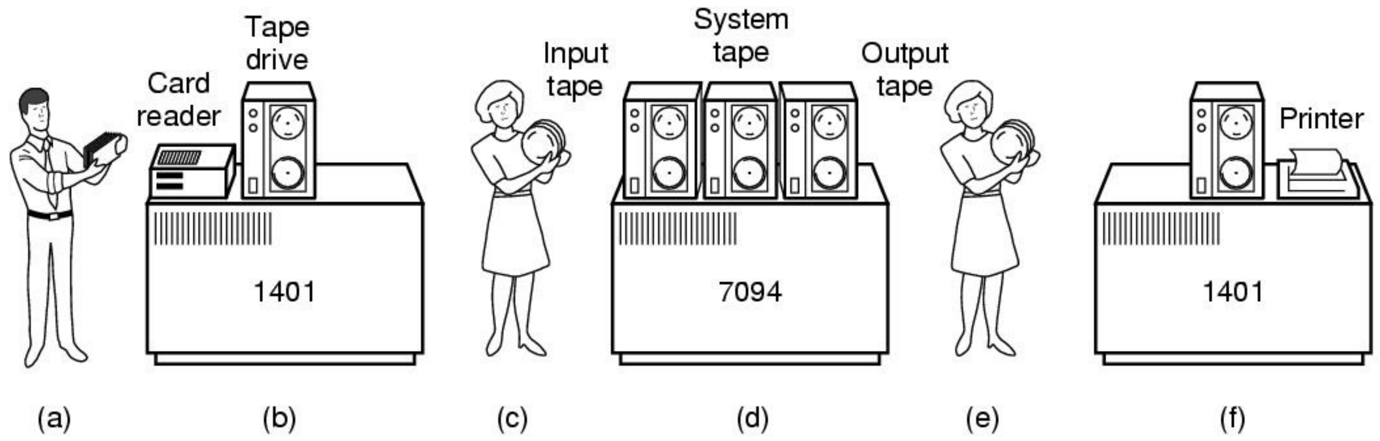


```
407M(1,J)=M(2,2*J-1):410NEXTJ:420FORJ=1TO100:425T(2,J)=0:426S(2,J)=0:427M(2,J)=0
```

In the Beginning…

- **There was no OS – just libraries**
  - Computer only ran one program at a time, so no need for an OS
- **And then there were batch systems**
  - Programs printed on stacks of punchhole cards
  - OS was resident in a portion of machine memory
  - When previous program was finished, OS loaded next program to run





In the Beginning…

- **There was no OS – just libraries**
  - Computer only ran one program at a time, so no need for an OS
- **And then there were batch systems**
  - Programs printed on stacks of punchhole cards
  - OS was resident in a portion of machine memory
  - When previous program was finished, OS loaded next program to run
- **Disk spooling**
  - Disks were much read stack onto disk while previous program is running
  - With multiple programs on disk, need to decide which to run next!
  - But, CPU still idle while program accesses a peripheral (e.g., tape or disk!)

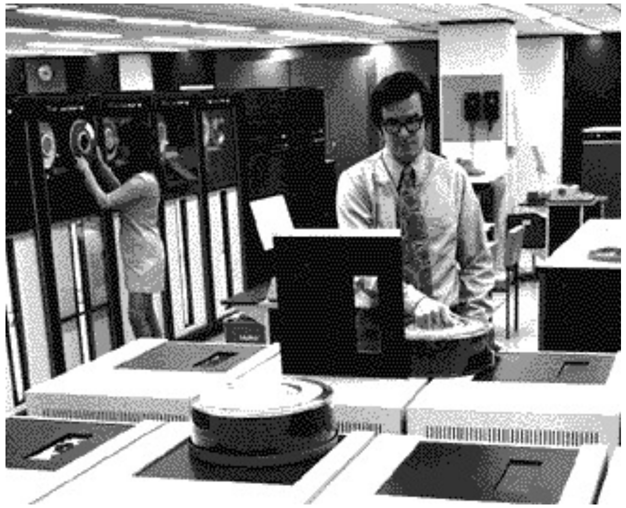
Multiprogramming

- **To increase system utilization, multiprogramming OS's were invented**
  - keeps multiple runnable jobs loaded in memory at once
- **Overlaps I/O of a job with computing of another**
  - While one job waits for I/O to compile, CPU runs instructions from another job
- **To benefit, need asynchronous I/O devices**
  - need some way to know when devices are done performing I/O
- **Goal: optimize system throughput**
  - perhaps at the cost of response time…

Dennis Ritchie and Ken Thompson at a PDP11, 1971

Timesharing

- **To support interactive use, timesharing OS's were created**
  - multiple terminals connected to one machine
  - each user has illusion of entire machine to him/herself
  - optimize response time, perhaps at the cost of throughput
- **Timeslicing**
  - divide CPU fairly among the users
  - if job is truly interactive (e.g. editor), then can switch between programs and users faster than users can generate load
- **MIT Multics (mid-1960's) was the first large timeshared system**
  - nearly all modern OS concepts can be traced back to Multics



Personal Computing

- Totally changed the computing industry.
- CP/M: First personal computer OS
  - IBM needed OS for their PCs, CP/M behind schedule
  - Bill Gates to the rescue: Bought 86-DOS and made MS-DOS
    - DOS is basically a subroutine library!
- Many popular personal computers follow
  - Apple, Commodore, TRS-80, TI 99/4, Atari, etc...

Apple I, 1976

Apple LISA, 1983

IBM PC, 1981

Bill Gates and Paul Allen, c.1975

Photo courtesy of Microsoft Archives.

Commodore VIC-20

Apple Introduces the First Low Cost Microcomputer System with a Video Terminal and 8K Bytes of RAM on a Single PC Card.

The Apple Computer. A truly complete microcomputer system on a single PC board. Based on the MOS Technology 6502 microprocessor, the Apple also has a built-in video terminal and sockets for 8K bytes of on-board RAM memory. With the addition of a keyboard and video monitor, you'll have an extremely powerful computer system that can be used for anything from developing programs to playing games or running BASIC.

Combining the computer, video terminal and dynamic memory on a single board has resulted in a large reduction in chip count, which means more reliability and lowered cost. Since the Apple comes fully assembled, tested & burned-in and has a complete power supply on-board, initial set-up is essentially "hassle free" and you can be running within minutes. At $666.66 (including 4K bytes RAM!) it opens many new possibilities for users and systems manufacturers.

You Don't Need an Expensive Teletype.

Using the built-in video terminal and keyboard interface, you avoid all the expense, noise and maintenance associated with a teletype. And the Apple video terminal is six times faster than a teletype, which means more throughput and less waiting. The Apple connects directly to a video monitor (or home TV with an inexpensive RF modulator) and displays 960 easy to read characters in 24 rows of 40 characters per line with automatic scrolling. The video display section contains its own 1K bytes of memory, so all the RAM memory is available for user programs. And the Keyboard Interface lets you use almost any ASCII-encoded keyboard.

The Apple Computer makes it possible for many people with limited budgets to step up to a video terminal as an I/O device for their computer.

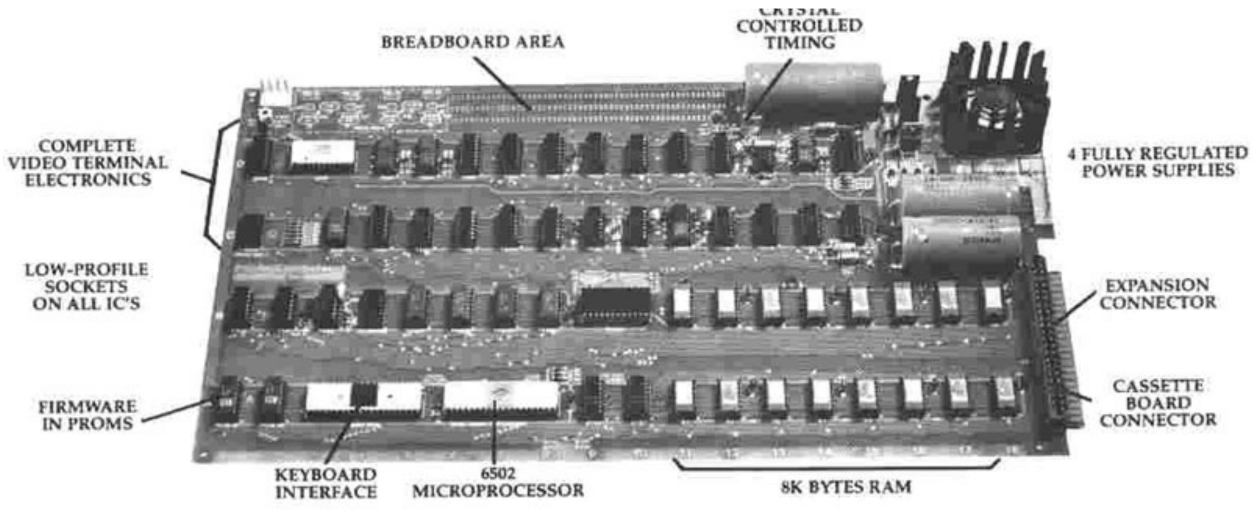
No More Switches, No More Lights.

Compared to switches and LED's, a video terminal can display vast amounts of information simultaneously. The Apple video terminal can display the contents of 192 memory locations at once on the screen. And the firmware in PROMS enables you to enter, display and debug programs (all in hex) from the keyboard, rendering a front panel unnecessary. The firmware also allows your programs to print characters on the display, and since you'll be looking at letters and numbers instead of just LED's, the door is open to all kinds of alphanumeric software (i.e., Games and BASIC).

8K Bytes RAM in 16 Chips!

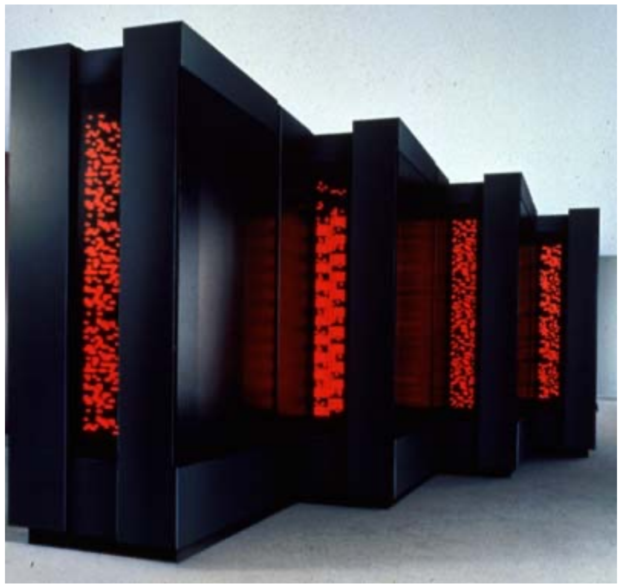
The Apple Computer uses the new 16-pin 4K dynamic memory chips. They are faster and take ¼ the space and power of even the low power 2102's (the memory chip that everyone else uses). That means 8K bytes in sixteen chips. It also means no more 28 amp power supplies.

The system is fully expandable to 65K via an edge connector which carries both the address and data busses, power supplies and all timing signals. All dynamic memory refreshing for both on and off-board memory is done automatically. Also, the Apple Computer can be upgraded to use the 16K chips when they become available. That's 32K bytes on-board RAM in 16 IC's —the equivalent of 256 2102's!



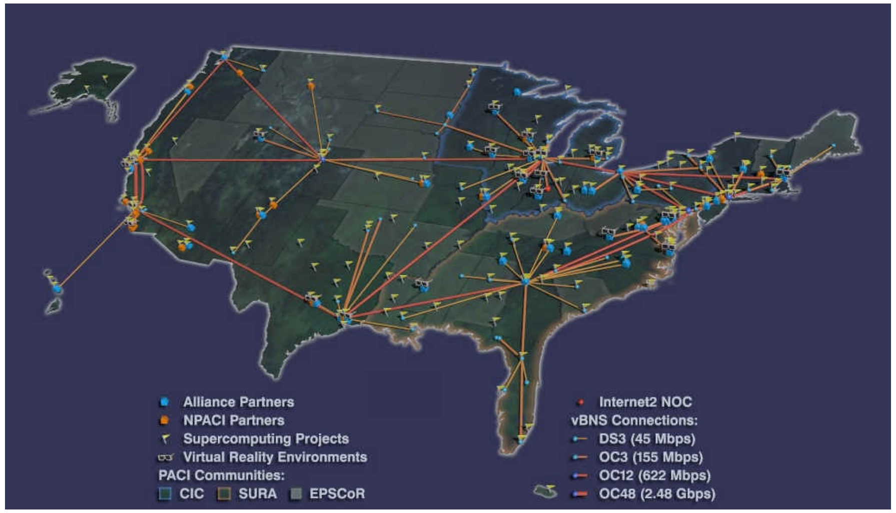
Parallel Computing and Clusters

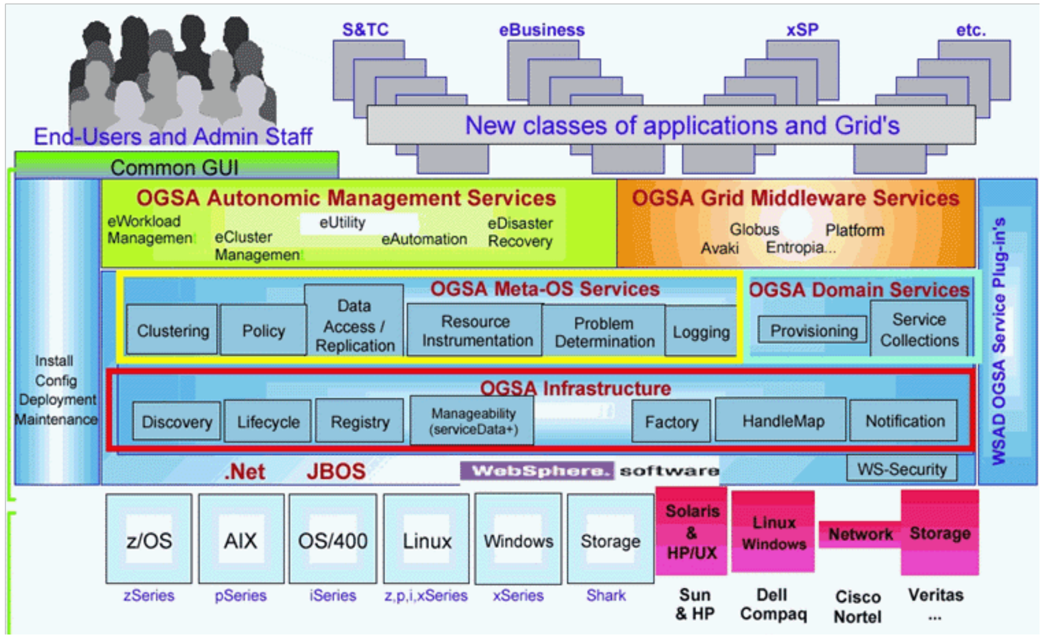
- **High-end scientific apps want to use many CPUs at once**
  - Parallel processing to crunch on enormous data sets
  - Need OS and language primitives for dividing program into parallel activities
  - Need OS primitives for fast communication between processors
    - degree of speedup dictated by communication/computation ratio
- **Many kinds of parallel machines:**
  - SMPs: symmetric multiprocessors – several CPUs accessing the same memory
  - MPPs: massively parallel processors – each CPU may have its own memory
  - Clusters: connect a lot of commodity machines with a fast network



Distributed OS

- **Goal – Make use of geographically distributed resources**
  - workstations on a LAN
  - servers across the Internet
- **Supports communication between applications**
  - interprocess communication (on a single machine):
    - message passing and shared memory
  - networking procotols (across multiple machines):
    - TCP/IP, Java RMI, .NET SOAP
- **“The Grid”, .NET, and OGSA**
  - Idea: Seamlessly connect vast computational resources across the Internet





Embedded OS

- **The rise of tiny computers everywhere – ubiquitous computing**
  - Processor cost low enough to embed in many devices
    - Cell phones
  - How many CPUs are in your car? On your body right now?
- **Gets more interesting with ubiquitous networking!**
  - Wireless networks is pervasive
  - Sensor networks are an exciting new direction here
    - Little "motes" with less 4KB of RAM, some sensors, and a radio
- **Typically very constrained hardware resources**
  - slow processors
  - very small amount of memory (e.g. 8 MB)
  - no disk – but maybe quasi-permanent storage such as EEPROM