ECE437/CS481

INTRODUCTION TO OS

OS Development & Evolution

Xiang Sun

The University of New Mexico

OS/Computer Evolution

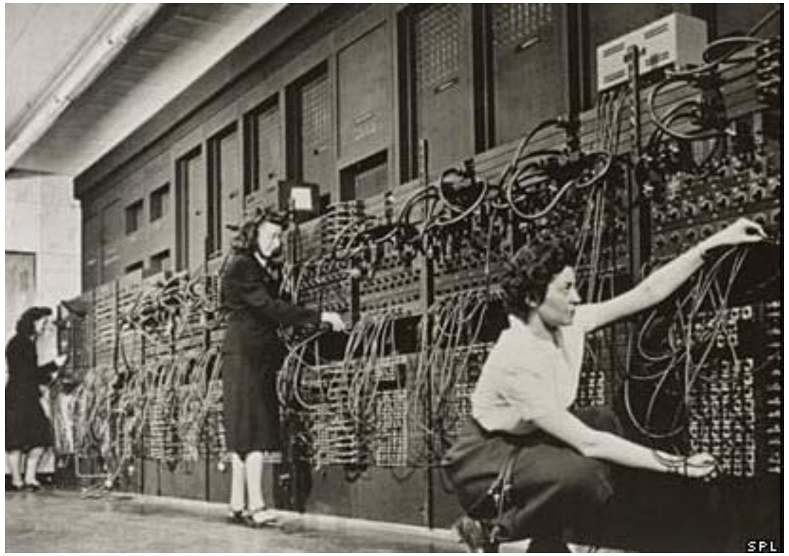
❑ Serial processing (1940s)

- manual loading and then execution
- manual operations on a bare machine--punch card, paper tapes, etc.
- hardware: vacuum tubes
- problems: inefficient use of the very expensive hardware

❑ Electronic Numerical Integrator And Computer (ENIAC)

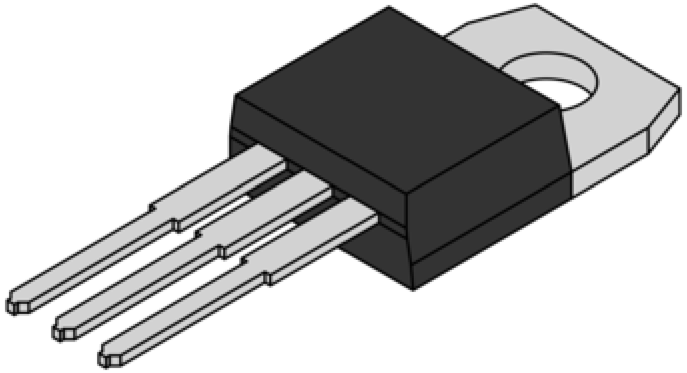
---Built at the U of Pennsylvania

- First general-purpose digital computer
- Weighted 30 tons; equipped 18,000 vacuum tubes
- A team of five operators working several days on the external wiring
- No operating system



OS/Computer Evolution

- Serial processing (1950s)
  - hardware: vacuum tubes → transistors
  - Program concept:
    - Programs to be stored & reload
    - Programs to be reused as subroutine calls



- John van Neumann Architecture

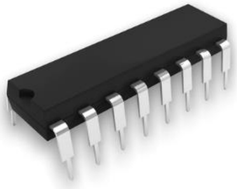
Programs and data could be represented in a similar way and stored in the same internal memory.

- IBM dominated the data processing industry
  - IBM 7070: first commercial transistorized computers
  - IBM 7090: 1) a 36-bit scientific machine; 2) with IBSYS operating system—a tape-based operating system

© by Dr. X. Sun

OS/Computer Evolution

❑ Batch processing (1960s)



- ➢ Hardware: transistors → ICs
- ➢ Batch processing OSs: collect the jobs (programs and data) together in a batch before processing starts
  - ✓ Automate the sequence of operations
  - ✓ Introduce Job Control Language (JCL) to instruct the system on how to run batch jobs
  - ✓ Introduce batch monitor

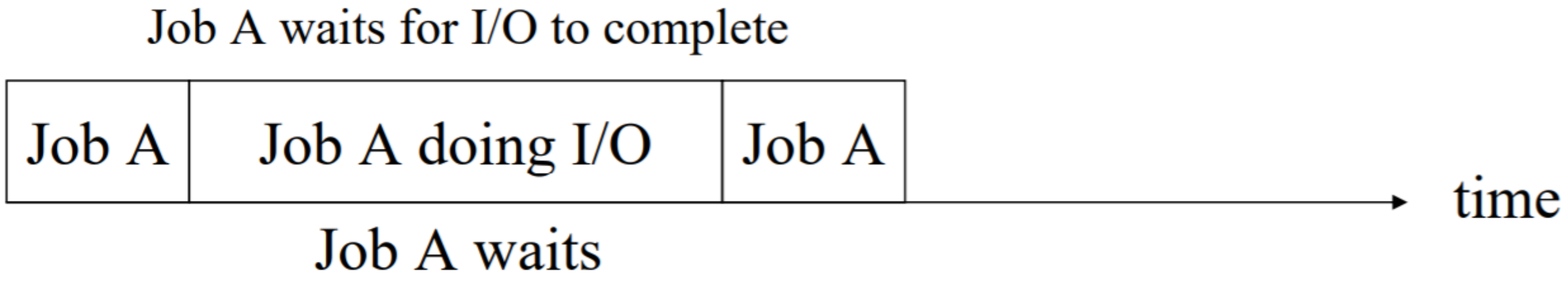
❑ IBM System/360 (S/360)

- ➢ It used microcode to implement the instruction set, which featured 8-bit byte addressing and binary, decimal and floating-point calculations.

OS/Computer Evolution

❑ Batch processing (1960s)

- Problems: sequential execution; that is, no interaction and no overlap between a fast CPU and slow I/O devices.



- Solutions:
  - ✓ I/O channel/buffering -- overlap I/O of a job with its own computation
  - ✓ SPOOL (simultaneous peripheral operation on-line) -- overlap the I/O of a job with other job's computation
  - ✓ Interrupts -- I/O devices can send signals to the CPU to get attention.

OS/Computer Evolution

❑ Time Sharing (1960s-1970s)

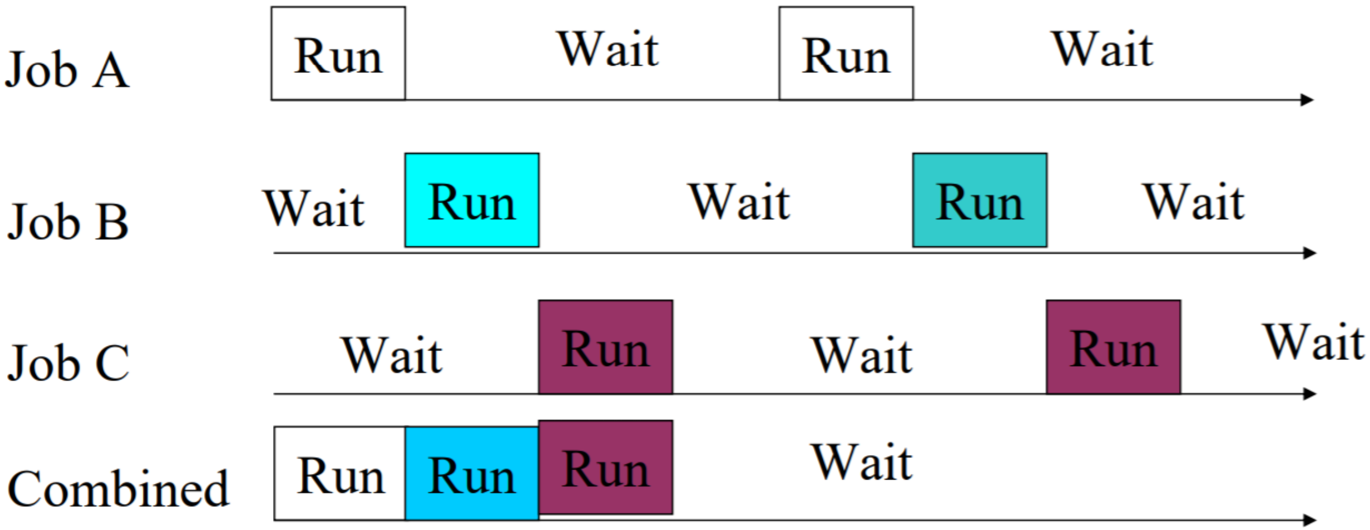
- Computing resources are assigned to different users (which are coming from different terminals) for a short time period, and thus a user in a terminal gets the feeling that he/she has dedicated computing resources to him/her behind her terminal.

- CTSS (Compatible Time-Sharing System)--One of the first time-sharing OSs
  - ✓ System clock generates interrupts at a rate of approximately one every 0.2 sec.
  - ✓ At each interrupt, OS could assign the processor to another user.
  - ✓ At regular time intervals, the current user would be preempted and another user loaded in.
  - ✓ Old user programs and data are written out to disk, and will be restored in main memory when that program was next given a turn.

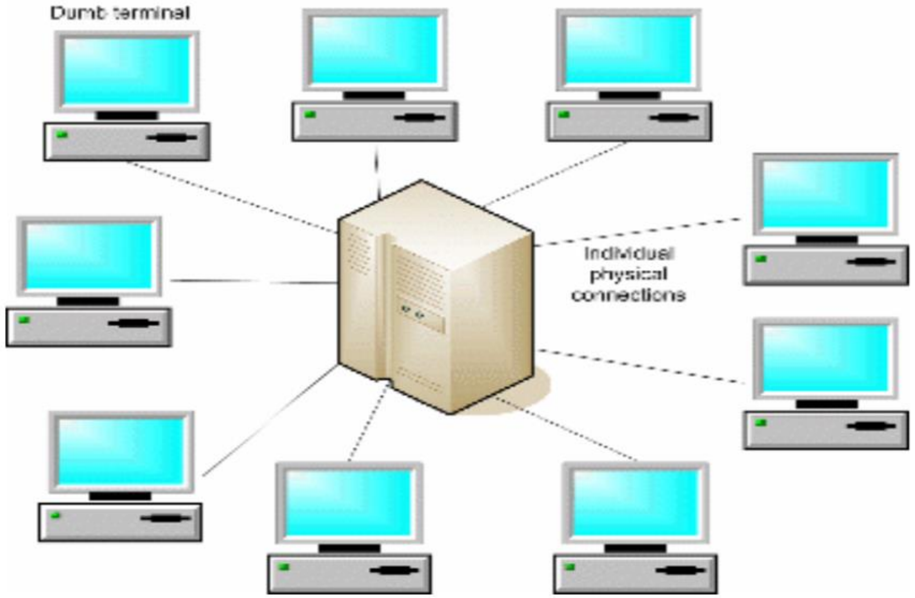
OS/Computer Evolution

❑ Multiprogramming (1970s)

- Having more than one program/jobs in memory at the same time.



Multiprogramming/Multitasking



Time Sharing

OS/Computer Evolution

❑ Multiprogramming (1970s)

- Differences between multiprogramming OSs and time sharing OSs.

Multiprogramming	Time Sharing
Allow multiple jobs to share resources	Allow multiple users to share resources
Jobs should be in the memory	Users should be in different terminals
Goal: to use resource efficiently (i.e., maximize the resource utilization)	Goal: providing a method to fairly share resource among users

OS/Computer Evolution

Question

- There are three types of jobs, i.e., Job A, Job B, and Job C, in the queue. The capacity of CPU is 1 MIPS.

	Job A	Job B	Job C
Number of instructions	100 instructions	1000 instructions	500 instructions
I/O time	1100 us	1000 us	700 us
CPU time	$100/1\times10^6$ sec=100 us	$1000/1\times10^6$ sec=1000 us	$500/1\times10^6$ sec=500 us
CPU utilization for a single job	100/(1100+100)=1/12	1000/(1000+1000)=1/2	500/(500+700)=5/12

- What are the overall CPU utilization for using simple batching processing and multiprogramming, respectively?

OS/Computer Evolution

- Answer
  - Simple Batching Processing
    - ✓ CPU Utilization =sum(CPU time)/(sum(CPU time)+sum(IO time))
      =4/11
  - Multiprogramming
    - Optimal CPU Utilization = sum(CPU Utilization)= 100%
  - What if CPU Utilization > 100%

OS/Computer Evolution

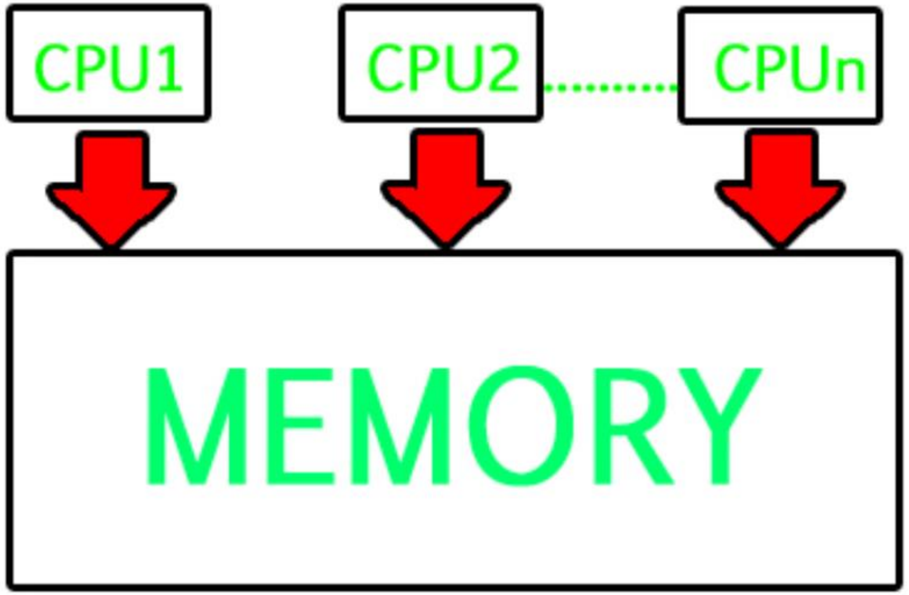
❑ Multiprogramming (1970s)

- ➢ Typical machines: IBM 360/370, PDP-7/11, Intel 8080
- ➢ Build the foundation of OS
  - ✓ The decision on which job to execute next from a queue/pool of ready jobs involves CPU scheduling.
  - ✓ Having several jobs ready to run implies that they must reside in memory, which requires Memory management.
  - ✓ Jobs may have to be swapped in/out of main memory to the disk implies that Disk management must be provided.
  - ✓ Multiple jobs running implies that OS must minimize the impact of one job on another, which introduces Protection.
- ➢ Birth of UNIX---written in a high level programming language.

OS/Computer Evolution

❑ Multiprocessing

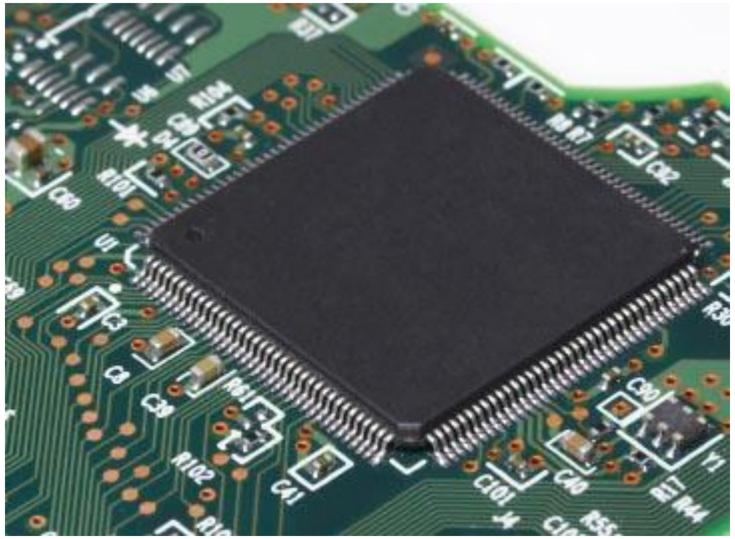
- ➢ Different processes can be assigned to different processors (cores) for their execution.
- ➢ Multiprocessing refers to the hardware (i.e., the CPU units) rather than the software (i.e., operating systems).
- ➢ A System can be both multi programmed by having multiple programs running at the same time and multiprocessing by having more than one physical processor.



© by Dr. X. Sun

OS/Computer Evolution

- ❑ Personal computers (1980s)
  - ➢ Hardware: ICs → LSI → VLSI →ULSI
  - ➢ Introduce microcomputers
  - ➢ Introduce GUI (Graphical User Interface) for OSs
  - ➢ Birth of MS-DOS
  - ➢ Typical machines: Intel 80286, IBM PC, Macintosh



OS/Computer Evolution

❑ Summary

- ➢ 1st generation OS--Serial Processing OS
  - ✓ Manual operations on a bare machine
- ➢ 2nd generation OS—Batch Processing OS
  - ✓ Introduce Job Control Language (JCL) to instruct the system on how to automatically run batch jobs
- ➢ 3rd generation OS—Time Sharing OS and Multiprogramming OS
  - ✓ Sharing resources among jobs/users
- ➢ 4th generation OS —OS on PCs
  - ✓ Introduce GUI