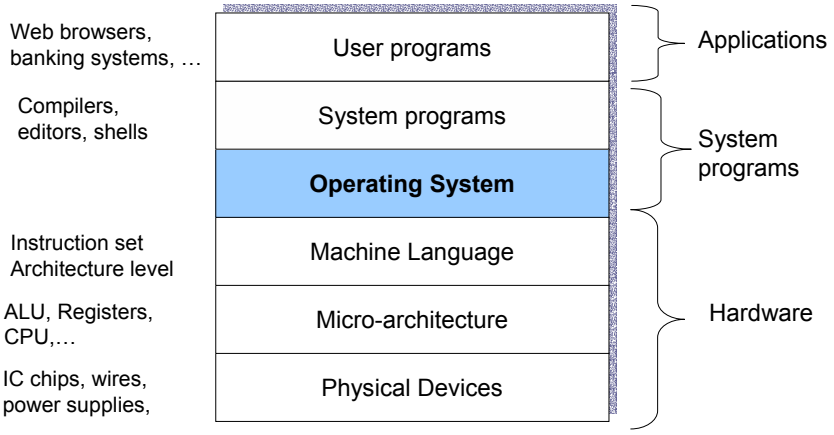
Introduction to Computer Systems and Operating Systems

CS 342 – Operating Systems
İbrahim Körpeoğlu
Bilkent University
Computer Engineering Department

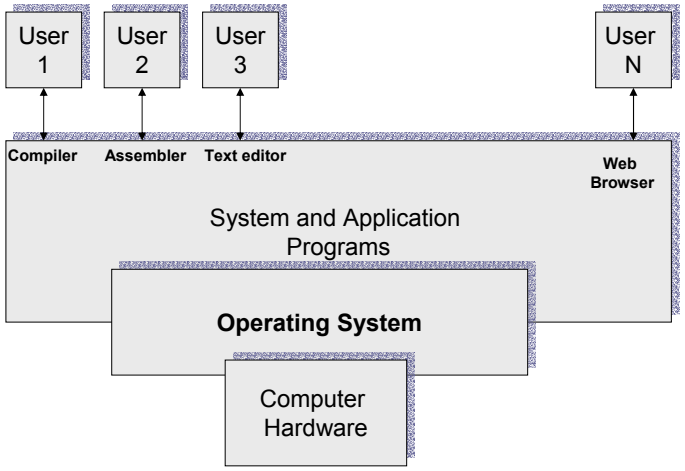
What is an Operating System

- An Operating System is a software
  - that lies between applications (programs) and hardware
  - that manages all the devices of a computer
  - that provides a convenient programming interface to the programs to access and use the hardware.

Placement of OS



Abstract View of components of computer system



Operating System from users and systems view

- Users view OS as a convenient interface to the hardware
  - OS provides system calls (OS level functions) that can be used by user applications to access the OS services.
  - OS shields the details of hardware from applications.
  - In this way, an OS is an extended machine or virtual machine that is easier to program for users to access and use the hardware.
- From computer systems perspective, OS is manager of hardware and software resources.

OS as extended machine

- The architecture of a computer is difficult to program
  - Architecture is: instruction set, memory organization, I/O, bus structure, etc.
  - To do some I/O work, a lot of hardware dependent work has to be done.
    - Example: to read from a floppy or hard disk:
      - One has to write the commands and addresses to the disk controller registers and then initiate the I/O
      - The disk controller will find the requested data in the disk and fetch it from disk to disk controller buffer.
      - One has to check the status of the disk controller operation whether it has finished or not.
      - If success, the data from the disk controller buffer should be moved to the main memory (to the application buffer)
- If all user programs would do these messy details:
  - The programs will be very difficult to write and they will be quite long.
  - The programs will be hardware dependent.
- Therefore, an OS cares all these details and provides a nice programming interface to the applications to access the devices.

OS as a resource manager

- There a a lot of resources in a computer system
  - CPU
  - Memory
  - I/O devices: network card, hard disk, serial port, USB port, floppy disk, printer, scanner, etc.
- If a computer system is used by multiple applications (or users), then they will compete for these resources.
- It is the job of the operating system to allocate these resources to the various applications so that:
  - The resources are allocated fairly
  - The resources are protected from cross-access
  - Access to the resources is synchronized so that operations are correct and consistent
  - Deadlocks are detected, resolved and avoided.

Goals for designing OSs

- Convenient service interface to user programs
  - System calls
  - Command interface
  - GUI
- Efficient resource allocation
  - Efficiency could be in terms of time, space, fairness, protection, security, etc.

- Usually, there are tradeoffs in designing an OS for a specific task (embedded systems, general purpose computers, mainframe computers).
  - Each will give more importance to some objectives than the others.
    - GUI is very important for pocket PC OSs
    - Efficiency is very important for mainframe OSs.

History of Operating Systems

- First generation (1945-55)
  - Systems that use vacuum tubes and plug-boards
  - No transistors
- Second generation (1955-1965)
  - Transistors used in these systems
  - Batch systems and mainframes
- Third generation (1965-1980)
  - ICs and Multiprogramming
- Fourth generation (1980 – present)
  - Personal Computers

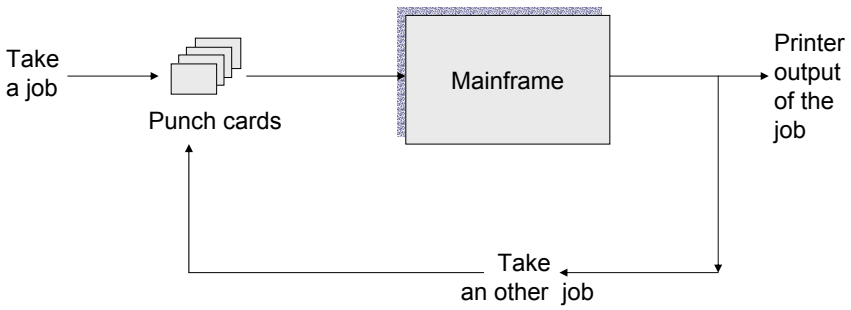
First generation

- Very big machines that fill up rooms.
- Very expensive.
- Designing, building, programming and operating the system was done by the same group of people.
- All programming is done in machine language.
  - By wiring up plug-boards (meaning direct access to the hardware). .
  - No programming languages
  - No Operating systems.
- Plug-boards are replaced later with punch-cards.
  - Programs are written on punch-cards and read by the machine.

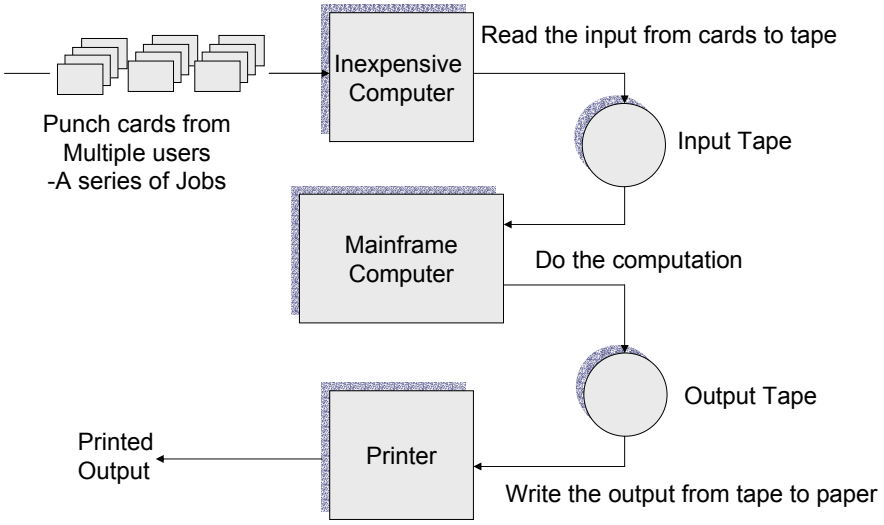
Second generation

- Transistors are used
- Machines become more reliable and faster
- Clean separation between designers, builders, operators and programmers.
- The machines that are produced are called mainframes.

Second generation – Early mainframes



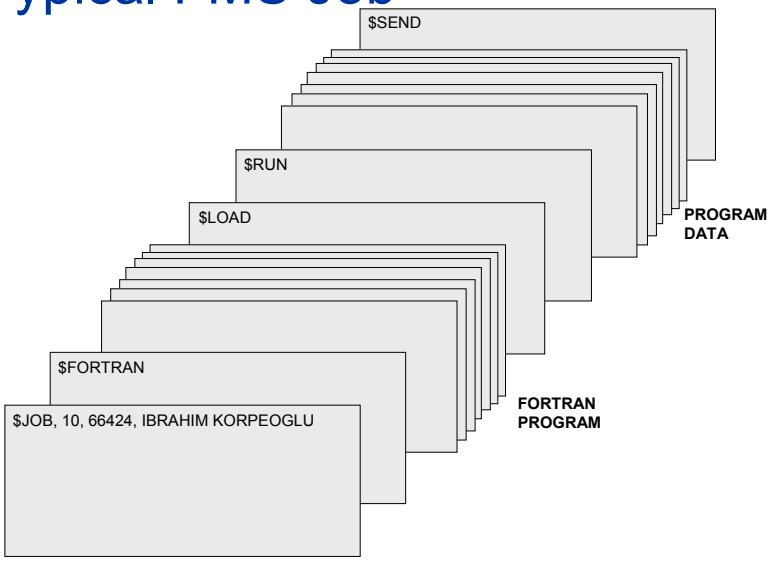
Second generation – Batch Sytems



Second generation – Batch Sytems

- The job of the OS was:
  - To load the next job (program) into main memory
  - To run the program with data
- An example system was IBM 7094.
- Typical OSs: FMS (Fortran Monitor System) and IBSYS
  - An OS was called monitor at that time.

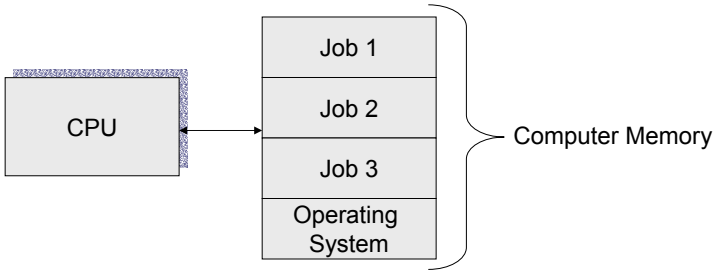
Typical FMS Job



Third Generation

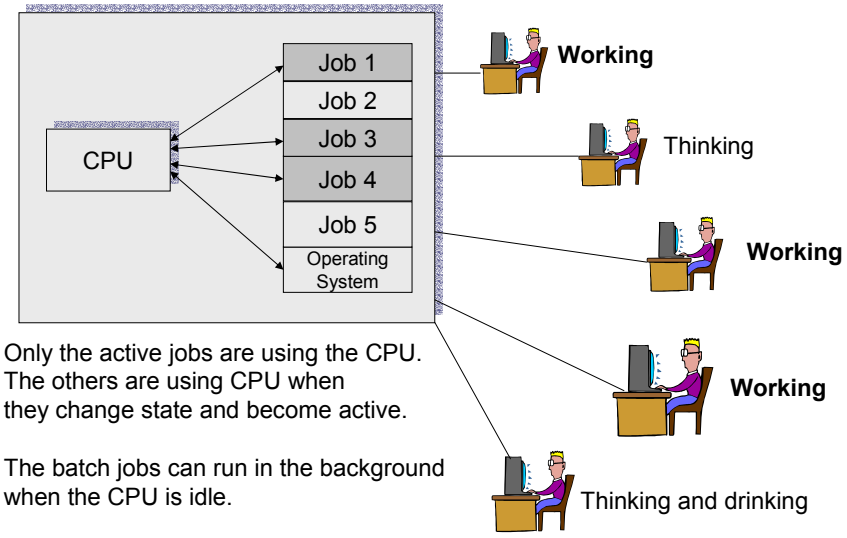
- Previously, computers were incompatible and they were designed for specific tasks:
  - Large-scale scientific computers
  - Commercial computers.
- IBM introduced System/360 family
  - A series of software compatible machines ranging from small (1401) to powerful (7094) ones.
  - They differed in configuration (memory size, CPU speed, etc) but not in architecture and instruction set.
  - Could be used for scientific and commercial needs.
  - They used Integrated circuits (small scale).
  - Very good price/performance.
- Required very sophisticated software (OS and applications) that should work on all members of the family.
- Multiprogramming is introduced with these computers.

Third Generation - Multiprogramming



- All jobs (programs) are loaded in memory
  - A program is not partially loaded, but the whole program is loaded.
  - Efficient CPU utilization.
  - CPU and memory is shared among multiple programs.
  - A job is removed from the memory when it has finished.
  - Requires protection of memory among different jobs.
- Only a limited number of jobs could be loaded into memory. The other jobs had to wait.
- This is still a batch system. Response time as very log. No user interaction during the execution of the job.

Third Generation – Time-sharing



Only the active jobs are using the CPU. The others are using CPU when they change state and become active.

The batch jobs can run in the background when the CPU is idle.

Third Generation – Time-sharing

- First time sharing system: CTSS developed at MIT in 1962.
  - CTSS: Compatible Time Sharing System
- Then MULTICS have been developed at MIT
  - It used many ideas that have been used by subsequent operating systems.
- Minicomputers have been also developed as part of 3rd generation systems
  - DEC PDP-1 (1961), which is following by PDP-11.
  - PDP-7 was the machine that Ken Thompson at Bell Labs used to implement the first Unix operating system.
  - Initial Unix is followed by System V Unix, BSD Unix, Linux,….

Fourth Generation – Personal Computers

- Large scale integrated circuits enabled development of machines with very small size and low price compared to the earlier ones.
- It became possible to buy computers for personal use.
- They are called personal or microcomputers.
- The design of the architecture was very similar to minicomputers or mainframes.

Fourth Generation – Personal Computers

- In early 1980s, IBM designed IBM PCs.
- MS-DOS was chosen as the operating system for IBM PC.
- GUI was used first by Xerox PARC researchers.
- Apple adapted GUI from Xerox in their computers. Apple later developed Macintosh.
- Microsoft got influenced from success of Macintosh and produced Windows.
- Windows initially was just a GUI system
- Later, with Win95, it was incorporated within operating system.
- Windows 98, Windows NT, Windows 2000, Windows ME and Windows XP followed.

Fourth Generation – Personal Computers

- Unix is run also in personal computers
  - FreeBSD, Linux, Solaris are examples
- X-windows system is developed at MIT and used as the GUI system of Unix operating systems
  - X-Windows system is completely a user level application, no operating system features are implemented inside X-windows.
- Network and distributed operating systems are developed also
  - Network operating system users are aware that there are multiple computers sharing a network and users explicitly login and use these computers.
  - Distributed operating system users are not aware that there are multiple computers available to them to run their programs. The system appears to them as if it is a uni-processor operating system.

Operating System Zoo

- There are a wide variety of operating systems depending on the hardware system that they are running on and depending on the need of users and depending on the target application class.
  - Mainframe operating systems
    - They have huge I/O capacity.
    - Very efficient resource utilization
    - Poor GUI.
    - Three type of jobs: batch, transaction processing, time-sharing.
    - Examle: OS/390

Operating System Zoo

- Server Operating Systems
  - Run on servers: very large personal computers, workstations, or mainframes.
  - Allows the sharing of hardware and software resources.
  - Services include: print service, web service, file service.
  - Examples: Unix and Windows 2000.
- Multiprocessor operating systems
  - Run on machines that have multiple processors running concurrently.
  - Need for communication and connectivity.
  - These machines are called multi-computers, parallel computers, or multiprocessors.

Operating System Zoo

- Personal Computer Operating Systems
  - Provides good GUI.
  - Resource management and protection is less stringent if there is a single user of a computer.
  - Examples: Windows 98, Windows 2000, Macintosh OS, Linux.
- Real-Time operating Systems
  - They have time as the key parameter.
  - An operation should complete before a well-specified deadline.
  - There are two kinds:
    - Hard-real time systems: a system that control robot arms in assembly lines. Miss of deadlines is intolerable.
    - Soft-real time systems: multimedia processing systems. They can miss deadlines sometimes without affecting the functionality. Example: VxWorks

Operating System Zoo

- Embedded operating systems
  - Embedded OSs runs on systems that control devices.
  - The systems usually may not have advanced user interface and interaction.
  - TV sets, microware ovens, mobile telephones, pocket PCs,
    - The system usually have very low memory, CPU, screen, etc.
  - Examples: PalmOS, Windows CE.
  - There are operating systems that are both embedded and real-time: The OSs for Internet routers and switches, for example.

Next

- Computer Hardware Review
  - What is a computer system
  - How does it works
  - What are components
  - How they are interconnected.
- Operating System Review
  - Structure of OS
  - Components of OS
  - Services that OS provides
    - System calls.

Course discussion

- Questions to ask
  - Are slides helpful in this course or whould you prefer writing on the blackboard?
    - Pros and cons of slides
  - Projects
    - Does everybody know C?
    - How many people is familier with unix?
    - Does everybody have unix account?
    - How many people have PC?
    - How many people have Unix on PC?
      - Number of Linux users?
      - Number of FreeBSD users?

Course discussion

- Things to stress on
  - Textbook should be read in a timely manner
    - Every week you should be upto date with the course
    - Don't leave the reading before the exam.
    - Pros:
      - you will enjoy reading,
      - you will understand better,
      - You will feel much less stress during exam period
  - Review the slides as soon as possible
  - Ask questions
    - In the class
    - Out of class
      - "Bilmemek değil Ögrenmemek kotüdür"

Course discussion

- Things to stress on
  - Do the homeworks
    - You will learn for life-long if you put enough effort for doing the homeoworks and understanding the topics
    - Doing homeworks will also be very helpful for exams.
  - Do the programming yourself
    - Don't cheat.
    - Write programs and be persistent on debugging.
      - "Every bug that you debug will be an experience for you".
      - Experience is very important in programming.