Operating Systems (CSE531)
Lecture # 01

Manish Shrivastava

CVIT, IIIT Hyderabad

Course Content

- Introduction
- System Structures
- System Calls
- Virtual Machine
- Process Concepts
- Operation on Processes
- Inter-Process Communication
- Multi-threaded Programming
- CPU Scheduling
- I/O and Storage
- Process Synchronization
- Semaphores & Monitors
- Deadlocks
- Memory Management
- Virtual Memory
- Distributed Systems
- System Security
- Practical Real-world OS

Reference Material

- Books
  - ❖ **Operating System Concepts by Silberschatz, A., Galvin, P., Gagne, G., Addison-Wesley (8th or latest edition).**
  - ❖ Others
    - ✓ Operating systems by William Stallings, Prentice-Hall, 1998.
    - ✓ Operating Systems by Gary Nutt, Pearson Education
    - ✓ Operating Systems: A design-oriented approach by Charles Crowley, Tata McGraw-Hill, 1997.
    - ✓ Operating Systems: Concepts and Design by Milan Milenkovic, TATA McGRAW-HILL
    - ✓ Tanenbaum, A., Modern Operating Systems, Prentice-Hall, second edition, 2000.
- Course Website http://moodle.iiit.ac.in
- Online Courses/Tutorials and Research Papers
- Refer to course plan on moodle.

Expected Outcome

- Do not revisit but **re-establish** your basics
    - Go deeper
    - Better understand
    - Be more up-to-date

- We will ensure that both ***theoretical*** as well as ***practical*** aspects are learnt simultaneously.

- The project deliverables are expected to working code/prototype.

Grading Scheme

- Relative Grading.

- Detailed Grading Breakup (~ 46% Exams/Quiz + 54% Practical Learning):
  - 24% Course Project
  - 30% Assignments (6×5)

  - 20% Mid-semester Exams (10×2)
  - 20% Final Exam
  - 06% Quiz/Others

- Zero marks in assignment if a copying is detected
- Cooperation with TA's is expected
- Zero tolerance from institute's attendance policy

Lecture 01: Plan

- System
- Characteristics
- System Concepts
- System Components
- Recap of historical OS efforts

What is an Operating System?

Abstracts the computer hardware

- Hides the messy details of the underlying hardware
- Present users with an easy to use resource abstraction
- Extends or virtualizes the underlying machine

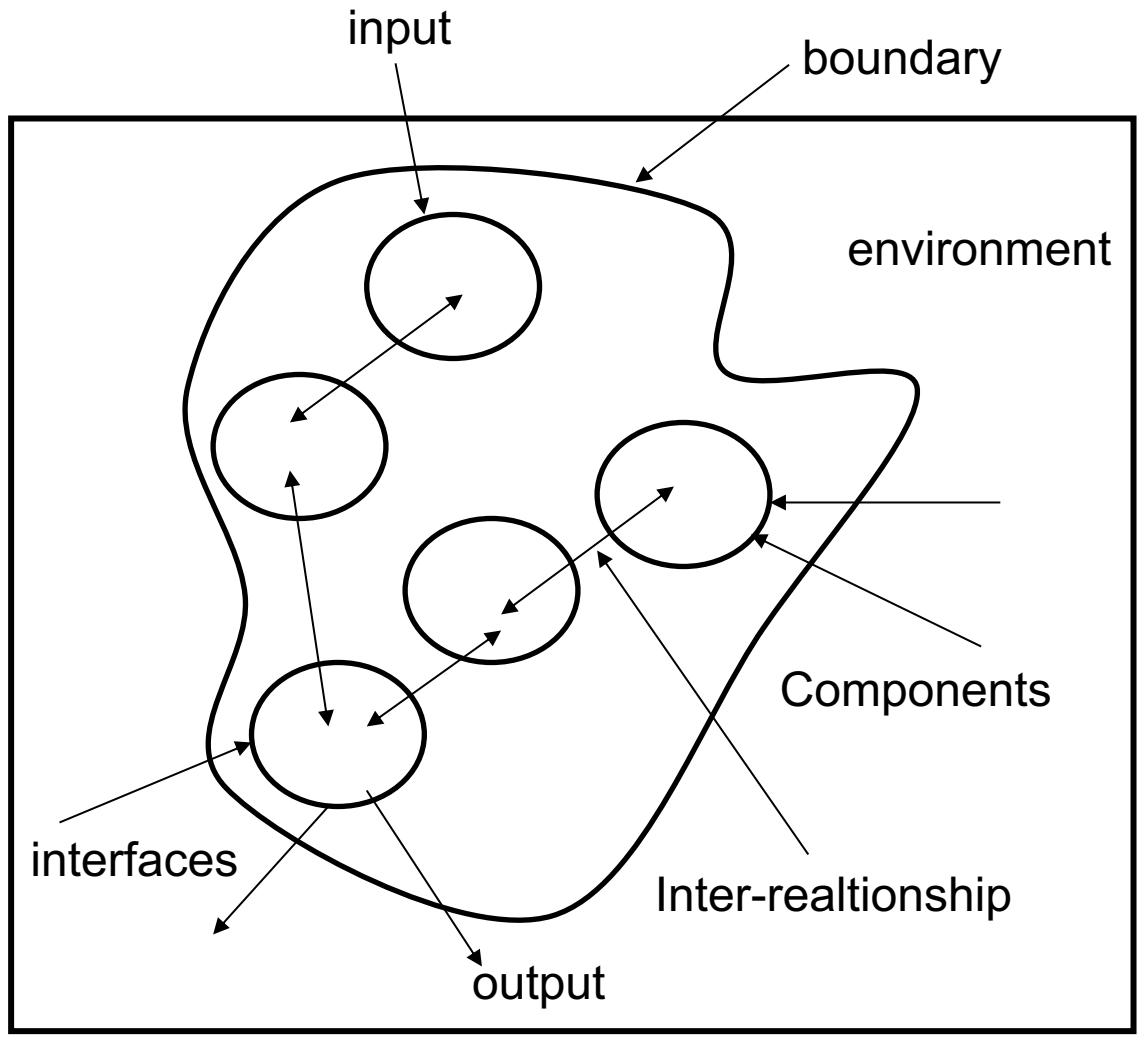
Manages the resources

- Processors, memory, timers, disks, mice, network interfaces, printers, displays, …
- Allows multiple users and programs to share the resources and coordinates the sharing

System

Characteristics of a system:

- Components
- Inter-related components
- A boundary
- A purpose
- An environment
- Interfaces
- Input
- Output
- Constraints



Characteristics

- Components:
  - A system is made up of components.
  - A component is either an irreducible part or aggregation of parts that make-up a system. A component is also called a sub-system.

- Interrelated Components:
  - The components of interrelated
  - Dependence of one subsystem on one or more subsystems.

Characteristics

- Boundary (Scope):
  - A system has a boundary, within which all of its components are contained and which establishes the limits of a separating the system from other systems.

- Purpose
  - The overall goal of function of a system. The system's reason for existing.

Characteristics

- Environment
  - Everything external to the system that interacts with the system.

- Interface
  - Point of contact where a system meets its environment or subsystems meet each other.

- Constraint:
  - A limit what a system can accomplish: Capacity, speed or capabilities.

Characteristics

- Input
  - Whatever a system takes from its environment in order to fulfill its purpose.

- Output:
  - Whatever a system returns to its environment in order to fulfill its purpose.

Important System Concepts

- Decomposition
- Modularity
- Coupling
- Cohesion

Decomposition

- Divide and Conquer Strategy
- It deals with being able to break down a system into its components.
- Decomposition results in smaller and less complex pieces that are easier to understand than larger, complex pieces.
- Decomposing a system also allows to focus on one particular part of a system, making easier to think of how to modify that part independently of the entire system.

Modularity

- Modularity refers to dividing a system up into chunks or modules of a relatively uniform size.
- You can replace or add any other module (or a component) without effecting the rest of the system.
- It is a design strategy in which system is composed of relatively small and autonomous routines fit together.

Coupling

- Coupling is the extent to which subsystems are dependent on each other.
- Subsystems should be as independent as possible.
- If a subsystem fails and other subsystems are highly dependent on it, the others will either fail themselves or have problems in functioning.

Cohesion

- The extent to which a system or a subsystem performs a single function.

Operating System (OS)

- The operating part of a tool is called as operating system of that tool.
- The purpose of operating system is to facilitate the operation of the underlying machine or tool.
- For a machine, the OS abstracts the machine part in terms of simple services by hiding the details of the machine.
- The OS can provide services to users or other subsystems.
- Examples of typical operating systems:
  - Car operating system, Telephone operating system, TV operating system and so on.

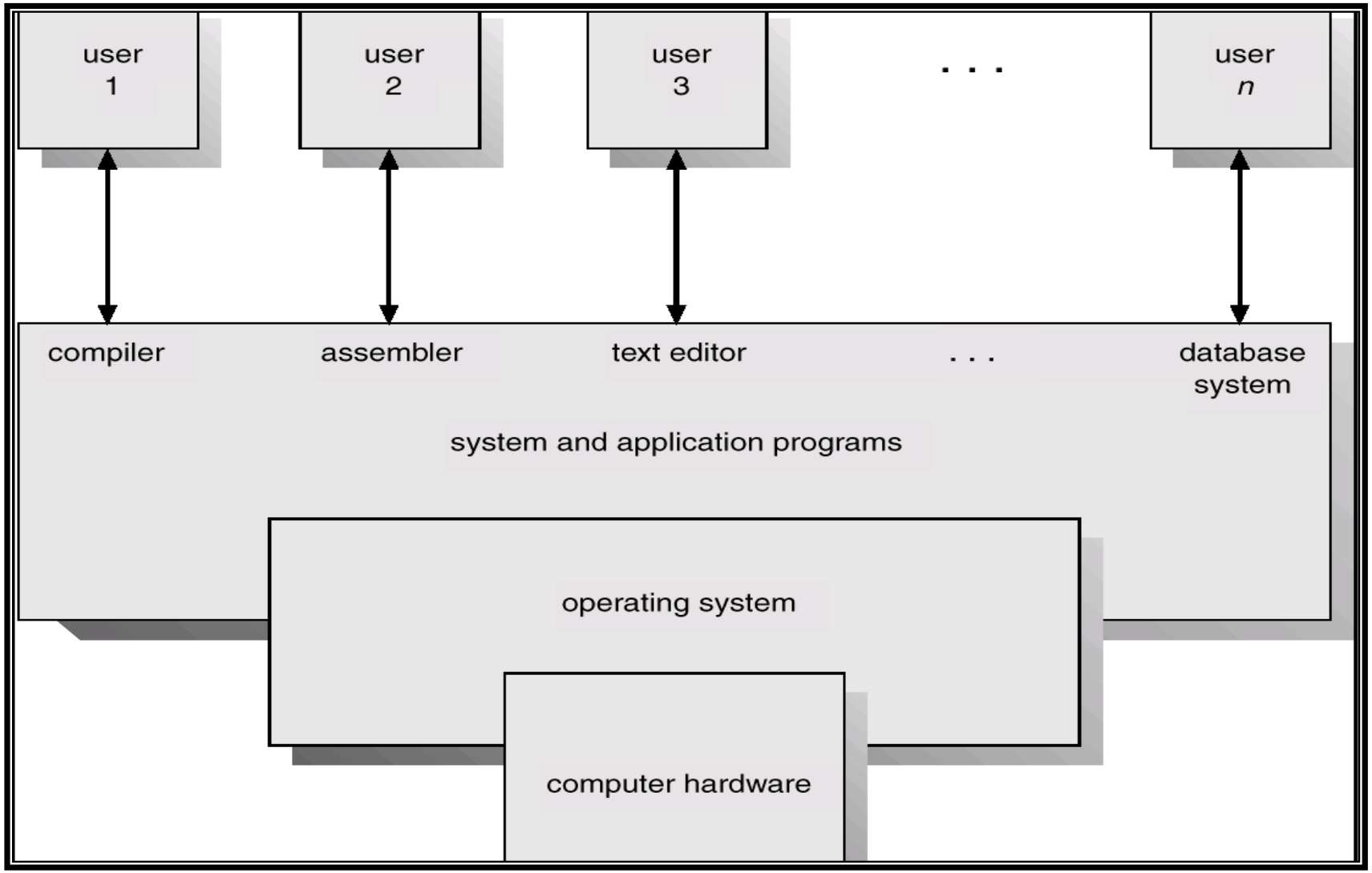
Computer Operating System

- A computer is also a tool that contains machine part and operating part.
- The operating part of a computer is called Computer Operating System.
- For a computer, the operating system abstracts the underlying hardware in terms of simple services by hiding the details of the hardware. The OS can provide services to users or other subsystems.
- Examples of Computer operating systems:
  - WINDOWS, Macintosh, UNIX, SOLARIS, LINUX and so on.
- In the rest of this course, operating system means computer operating system.

Computer OS components

- **Hardware** – provides basic computing resources (CPU, memory, I/O devices).
- **Operating system** – controls and coordinates the use of the hardware among the various application programs for the various users.
- **Applications programs** – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
- **Users** (people, machines, other computers).

Computer OS components



Other OS Definitions

- **Resource allocator** – manages and allocates resources.
  - Resources: CPU time, Memory Space, file storage space, I/O devices and son on.
- **Control program** – controls the execution of user programs and operations of I/O devices .
- **Kernel** – the one program running at all times (all else being application programs).
- The two goals, efficiency and convenience are sometimes contradictory.
- Much of OS theory concentrates on optimal use of resources.

Objectives

- The main objective is to understand the operational part of any computer.
- Understanding the general principles of OS design.
  - Focus on general-purpose, multi-user, uni-processor systems.
  - Emphasis on widely applicable concepts rather than any specific features of any specific OS.
- Understanding problems, solutions and design choices.
- Understanding the structure of specific OSs: UNIX, LINUX, WINDOWS

Early Systems (Serial Processing)

- **1940-50:**
  - The programmer interacted directly with the computer hardware.
  - Display light, switches, printer, card reader.
  - No OS. Error is displayed through lights.

- **Problems:**
  - Scheduling → Users spend lots of time at the computer.
    - Signup sheet was used.
  - Job Setup time
    - Loading and compiling
      - Mounting and Un-mounting of tapes
      - Setting up of card desks
  - Libraries of functions, linkers, loaders, debuggers, and I/O driver routines were available for all the users.

Early Systems

- Early computers were (physically) large machines run from a console.
- The programmer would operate the program directly from the console.
  - The program is loaded to the memory from panel of switches, paper tape, and from punched cards.
- As time went on, additional software and hardware were developed.
  - Card readers, line printers, and magnetic tape became common place.
  - Libraries, loaders, and common functions were created.
    - Software reusability.

IBM029

Early Systems

- The setup time was a real problem
- CPU is idle while tapes are being mounted or the programmer was operating the console.
- In the early days, few computers were available and they were expensive (millions of dollars).
  - operational costs: power, cooling, programmers.
- Main question:

How to increase the utilization of CPU ?

Early Systems

- The solution was two fold.
- First, a professional computer operator was hired.
  - Once the program was finished, the operator could start next job.
  - The operator sets up the job, produces the dump, and starts the next job.
  - The set up time was reduced due to operator's experience.
- Second, jobs with similar needs were batched together and run through the computer as a group.
  - For example, if there is a FORTRAN job, COBOL job, and FORTRAN job, two FORTRAN jobs were batched together.
- However, during transition time CPU sat idle.
- Automatic job sequencing to avoid CPU idle time.
  - A first rudimentary OS was created
  - A small program called a **resident monitor** was developed.
  - The resident monitor always resided in memory.

Simple Batch Systems

- In serial systems
  - Machines were very expensive
  - Wasting time was not acceptable.
- To improve usage, the concept of batch OS was developed.
- The main idea is the use of software known as monitor.
  - The user no longer has access to machine.
- The user submits the job (tape) to the operator.
- The operator batches the jobs together sequentially, places entire batch as an input device for use by the computer.

Features of Batch System

- The batch OS is simply a program.
- It relies on the ability of the processor to fetch instructions from various portions of main memory to seize and relinquish control.
- Hardware features:
  - **Memory protection**: While the user program is running, it must not alter the memory area containing the monitor.
  - If such is the case the processor hardware should detect the error and transfer control to monitor.
  - **Timer:** A timer is used to prevent the single job from monopolizing the system

Features of Batch System

- Hardware Features
  - **Privileged instructions**
    - Contains instructions that are only executed by monitor.
    - I/O instructions
    - If a program encounters them the control shifts through monitor.
  - **Interrupts:** It gives OS more flexibility.
    - Relinquishing control and regain control
- With batch OS the machine time alters between execution of user programs and execution of monitor.
- Two overheads
  - Machine time is consumed by the monitor.
  - Memory is consumed by the monitor.
- Still, they improved the performance over serial systems.

Problem with Batch System

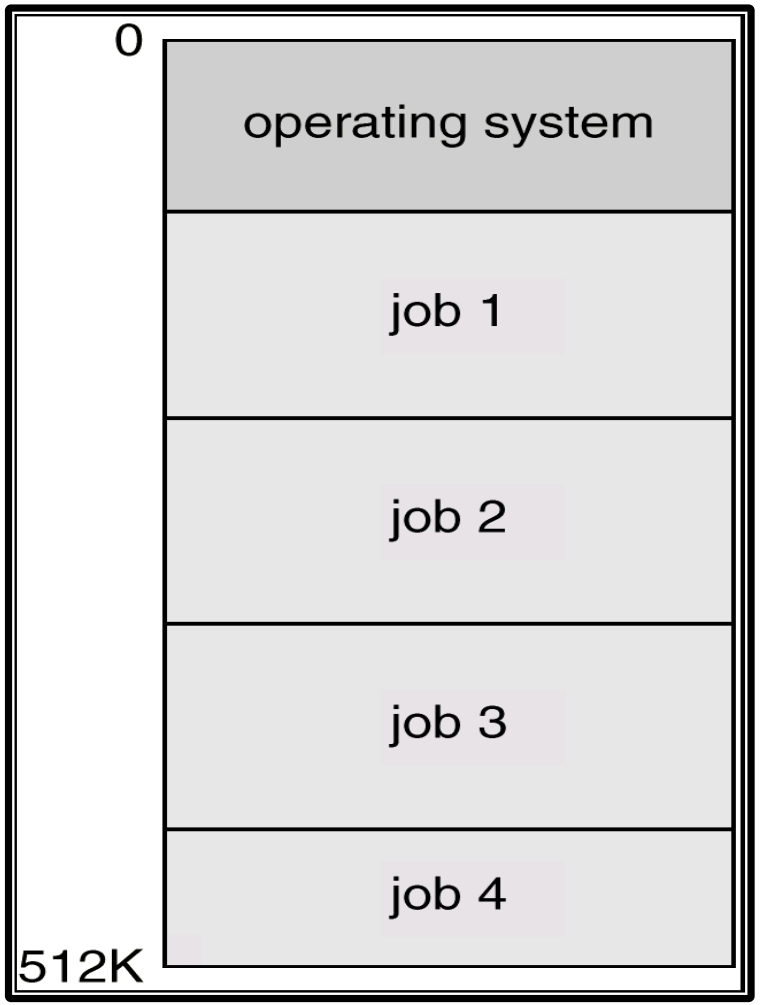
- CPU is idle
- Speed of mechanical devices is very slower than those of electronic devices.
- CPU works in a microsecond range
  - Thousands of instructions/second
- A card reader may read 1200 cards per minute (20 cards per second)
- CPU speed has increased at a faster rate.
- Tape technology improved the performance little-bit.
- Main perceived problem
  - Turn-around time: upto two days
  - CPU often underutilized
    - Most of the time was spent reading and writing from tape.

Multi-programmed Batched Systems (1960s) (or Multi tasking)

- If CPU is executing a job and requires a tape to be mounted
  - In a non multi-programmed system
    - CPU sits idle.
  - In a Multi-programmed system
    - CPU takes up another job.
- **Multiprogramming is the first instance when the OS started taking decisions.**
- Job scheduling is done by OS.
- Having several programs in the memory requires memory management.

Multi-programmed Batched Systems (1960s) (or Multi tasking)

- A single user can not keep either CPU or I/O busy.
- Multiprogramming increases CPU utilization by organizing jobs such that the CPU always has one to execute.
- The OS keeps several jobs in memory at a time and CPU is multiplexed among them



THANK YOU