INTRODUCTION TO OPERATING SYSTEMS
Definition of OS

- An operating system (OS) is the software that manages the sharing of the resources of a computer and provides programmers with an interface used to access those resources.
- Processes system data and user input, and responds by allocating and managing tasks and internal system resources as a service to users and programs of the system.
- Acts as an intermediary between the computer user and the computer hardware.
Variation in accomplishment of tasks

- Mainframe OS ➔ optimize utilization of Hardware
- PC OS ➔ complex games and business applications
- OS for handheld computer ➔ environment in which a user can easily interface with the computer
OS can be designed to be:

- Convenient
- Efficient
- Both
COMPUTER SYSTEM

- Hardware
- OS
- Application programs
- users
USER VIEW : varies according to interface being used

- **PC**  → ease of use
- **User at a terminal connected to a mainframe**  → maximize resource utilization
- **User at workstations connected to N/W’s of other workstations**  → individual usability and resource utilization
- **Handheld computers**  → individual usability
- **Embedded computers**  → no user view
System view: resource allocator

- Resources of a computer system:
  1. CPU time
  2. memory space
  3. file-storage space
  4. I/O devices

- Resource allocation is the process of allocating resources to specific programs and users so that it can operate the computer system efficiently and fairly

- Important where many users access the same mainframe
Tasks performed

- Controlling and allocating memory
- Prioritizing system requests
- Controlling i/p and o/p devices
- Facilitating networking
- Managing file systems
Computer system organization:

- One or more CPUs
- Number of device controllers

- Connected through a common bus that provides access to shared memory
- Can execute concurrently competing for memory cycles
Bootstrap program:

- A program that should be run for a computer to start running, for instance, when it is powered up or rebooted
- Stored in ROM or EEPROM
- Locate and load into memory the OS kernel
Interrupt mechanism:

- Interrupt must transfer control to the appropriate interrupt service routine.
- Interrupt routine is called indirectly through the table, with no intermediate routine needed.
- Interrupt architecture must also save the address of the interrupted instruction.
- After the interrupt is serviced, the saved return address is loaded into the program counter, and the interrupted computation resumes.
Functioning of I/O structure

- **Computer system:**
  1. CPUs
  2. Multiple device controller

- To start an I/O operation, the device driver loads the appropriate registers within the device controller.

- The device controller examines the contents of these registers to determine what action to take.
- Controller starts transfer of data from device to local buffer
- Once the transfer of data is complete, the device controller informs the device driver via an interrupt that it has finished its operation
- The device driver then returns control to OS
DMA

- After setting up buffers, pointers and counters for the I/O device, the device controller transfers an entire block of data directly to or from its own buffer storage to memory, with no intervention by the CPU.
- Only one interrupt per block is generated to tell the device driver that the operation has completed.
OS structure:

- **Multiprogramming**: increases CPU utilization by organizing jobs (code and data) so that the CPU has always one to execute.
- Drawback: does not provide for user interaction with computer system.
- **Timesharing**: (multitasking)
  - allows many users to share the computer simultaneously.
  - uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer.
- Process: a program loaded into memory and executing
- Job pool: job pool consists of all processes residing on disk awaiting allocation of memory
- Job scheduling: The ability of the OS to choose among several jobs that are ready to be brought into memory when there is no enough room for them
- CPU scheduling: The ability of the OS to choose among several jobs that are ready to run at the same time
Swapping: The technique in which processes are swapped in and out of main memory to the disk

Virtual memory technique: a technique that allows the execution of a process that is not completely in memory
- Trap: is a software generated interrupt caused either by an error or by a specific request from a user program that an OS service be performed
- In order to ensure the proper execution of the OS, we must be able to distinguish between the execution of OS code and user defined code
- Hardware support: differentiate between different modes
Dual mode operation

- User mode
- Kernel mode
Mode bit

- Added to the hardware of the computer to indicate the current mode
- Kernel: 0
- User: 1
- Distinguishes between a task that is executed on behalf of the OS and the one that is executed on behalf of the user
Protection accomplished

- By designating some machine instructions that may cause harm as privileged instructions
- The Hardware allows privileged instructions to be executed only in the kernel mode
- Hardware treats the instruction as a trap when an attempt is made to execute a privileged instruction in user mode
Timer

- To prevent a user program from getting stuck in an infinite loop or not calling system services and never returning control to the operating system
- Set to interrupt the computer after a specified period
- Period: fixed or variable
Variable timer

- Implemented by a fixed rate clock and a counter
- OS sets the counter
- On every tick, count is decremented
- When count=0, an interrupt occurs
Process management

- Process: defined as a program in execution
- Examples: 1. time shared user program such as a compiler
  2. a word processing program being run by an individual user on a PC
  3. system task like sending output to a printer
- Resources are given to a process either when it is created or while it is running.
- In addition to logical and physical resources, various initialization data may be passed along.
- When the process terminates, the OS claims back any reusable resources.
- Single threaded program: one program counter specifying the next instruction to execute
- CPU executes one instruction after the other until the process is complete
- Multi threaded program: multiple program counters, each pointing to the next instruction to execute for a given thread
Duties of OS in connection with process management:

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process communication
- Providing mechanisms for process synchronization
- Providing mechanisms for deadlock handling