

Lesson 3

Objectives

- Computer System Organization
- Operating System Components

COMPUTER SYSTEM ORGANIZATION

Computer-system operation

- One or more CPUs, device controllers connect through common bus providing access to shared memory
- Concurrent execution of CPUs and devices competing for memory cycles
- I/O devices and the CPU can execute concurrently.
- Each device controller is in charge of a particular device type.
- Each device controller has a local buffer.
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller.
- Device controller informs CPU that it has finished its operation by causing an *interrupt*.

Interrupt:

An interrupt is a signal from a device attached to a computer, or from a program within the computer, that tells the OS (operating system) to stop and decide what to do next. When an interrupt is generated, the OS saves its execution state by means of a context switch, a procedure that a computer processor follows to change from one task to another while ensuring that the tasks do not conflict. Once the OS has saved the execution state, it starts to execute the interrupt handler at the interrupt vector.)

OR

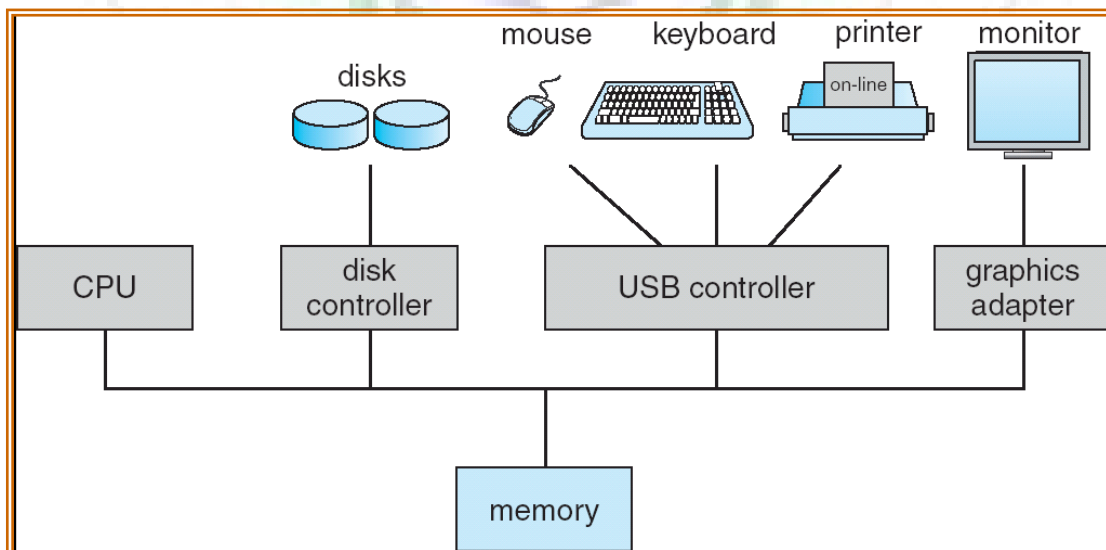
In digital computers, an interrupt is an input signal to the processor indicating an event that needs immediate attention. An interrupt signal alerts the processor and serves as a request for the processor to interrupt the currently executing code, so that the event can be processed in a timely manner.

Common functions of Interrupts:

- Interrupt transfers control to the interrupt service routine/interrupt handler generally, through the *interrupt vector*, which contains the addresses of all the service routines.
- Interrupt architecture must save the address of the interrupted instruction.
- Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt*.
- A *trap* is a software-generated interrupt caused either by an error or a user request.
- An operating system is *interrupt* driven.

Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter.
- Determines which type of interrupt has occurred:
 - *Polling* (Polling also refers to the situation where a I/O device is repeatedly checked for readiness, and if it is not, the computer returns to a different task.)
 - *vectored interrupt system* (An interrupt vector is the *memory* location of an interrupt handler, which prioritizes *interrupts* and saves them in a *queue* if more than one interrupt is waiting to be handled.
- Separate segments of code determine what action should be taken for each type of interrupt



OPERATING SYSTEM COMPONENTS

In order to get a complete overview of operating system we need to observe its individual components, their functionality and association with each other etc. these components are given subsequently.

1. Process Management

A program by itself is nothing unless its instruction executed by a CPU. A *process* is basically is “duration and steps of a program execution” or “unit of work in a system”.

Process
Executable Program (.EXE)
Program's data
Stack information, Stack pointer and registers involved in execution

Operating system maintains a process table for all the processes in terms of array of structures or linked lists. In time sharing systems, when more than one processes are simultaneously running sometime a process need to be freed from CPU, so kept in a suspended-mode. But before going to this mode its current status is stored in process table called *core image*, so that can be resumed afterwards. There are mainly two types of processes:

- System processes (OS services)
- User processes (user programs etc)

A process may contain sub-processes inside it, so they might give a tree view. Operating systems is capable of:

- Create/delete both system and users processes
- The suspension/resumption of system and user processes
- Provision of process synchronization
- Provision of process communication

2. Main memory management

CPU only deals with main memory (RAM). For example, if some program is needed to utilize CPU it has to go through by main memory similarly, if CPU produces some effects they went to main memory and then to disk drives etc. Similarly, the peripherals also deal with main memory then with CPU if needed.

Operating systems do following tasks:

- Keep track of which part of memory is currently being used and by whom?
- Decide which process is to be load in memory when space available
- Allocation/de-allocation of memory

3. File Management

The main structures of an operating system we mostly have to deal with are:

- File
 - i. It is a logical storage unit
 - ii. It is related collection of information defined by creator
 - iii. It consists of program (.exe, .bat, .com etc) or data (.txt etc)
 - iv. It is a sequence of bits/bytes/lines/ records
- Directories (Folders in windows)
 - i. These are logical partition (a named space)
 - ii. They may contain sub-directories and files
 - iii. Their hierarchy level could be considerably large
 - iv. A file location from its root directory to home directory called path
 - v. Directories don't occupies any space by themselves
 - vi. Directories are used to group the files.

The visible components of a file are:

- Capacity
- Data transfer rate
- Access method (sequential/ random block)

Operating system is to monitor:

- Creation /deletion of file and directories
- Mapping files onto to disk
- Backing up files on secondary media

- Privacy

4. I/O System Management

One of the main purposes of an OS is to provide an abstract version of hardware, means to display hardware in sense of some file or some named entity. For example, in windows control panel provide all hardware list in terms of directories, like printer, modem, scanner etc. Operating system manages these components by:

- Their memory management (buffering, caching, spooling etc)
- A general device driver interface
- Drivers for specific hardware device

5. Secondary storage management

- Free space management
- Storage allocation
- Disk scheduling (de-fragmentation)

6. Networking

One of the most elegant operations of operating system is networking. Where more than one computer systems (no matter what kind of they are) can communicate via a common bus of some other media. In this way operating system manages:

- Security risks
- Data transfer
- Layers management
- User authentication
- Session management and much more

7. Protection System

One of the key features of operating system is to handle concurrent execution of multiple processes of different users without any interference with each other. In other words, no inter user interference, nor inter-processes interference, also which resources are allowed to which particular user and his authentication, privileges maintenance etc.