**COMPUTER SYSTEM ORGANIZATIONS**

Before we explore the details of how computer systems operate, we need general knowledge of the structure of a computer system.

1. COMPUTER SYSTEM OPERATION

* 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 controller.
* Device controller informs CPU that it has finished its operation by causing an interrupt.
* Hardware may trigger an interrupt at any time by sending a signal to CPU.
* Software may trigger an interrupt by executing a special operation called system call/ memory call.

1. Interrupt Handling

When CPU is interrupted, it stops what it is doing and immediately transfers execution to a fixed location. Fixed location usually contains the starting address where the service routine for interrupt is located. The interrupt service routine executes; on completion.

Only a predefined number of interrupts are possible

1. STORAGE STRUCTURE

CPU can load instructions only from memory general purpose computers run most of their programs from rewritable memory called main memory (RAM). Main memory commonly is implemented in a semiconductor technology called dynamic random access memory (DRAM).

Ideally we want the programs and data to reside in main memory permanently. This arrangement usually is not possible for the following two reasons:

1. Main memory is usually too small to store all needed programs and data permanently.
2. Main memory is a volatile storage device that loses data when power is turned off.

Thus, computer systems provide secondary storage as an extension of main memory like Magnetic Disks (hdd), cache memory, CD-ROM.

**STORAGE SYSTEM HIERARCHY**

The main difference among various storage systems lie in speed, cost, size and volatility.

**Volatile DRAM &**

**Non-Volatile**

**Volatile**

**Non-Volatile**

The higher levels are expensive but they are fast. As we move down the hierarchy, the cost per bit generally decreases, whereas the access time generally increases.

**Storage systems are of two types Volatile and Non-Volatile.**

* VOLATILE STORAGE:

Loses its contents when power to the device is removed.

Example: registers, cache, main memory (RAM).

* NON-VOLATILE STORAGE:

It does not lose its contents when power of the device is removed.

Example: magnetic disk, optical disk, magnetic disk, magnetic tapes, ROM, flash memory.

It is typically used for secondary storage.

* VOLATILE STORAGE or NON-VOLATILE STORAGE

Electronic disk can be either volatile or non-volatile. It stores data in large DRAM array which is volatile. If external power is interrupted the electronic disk controller copies the data from RAM to magnetic disk because there is a hidden magnetic hard disk and a battery backup.

* FLASH MEMORY (type of electronic disk)

Used in cameras, personal digital assistants (PAD’s), in robots and in general purpose computers for removable memory. Flash memory is slower than DRAM but needs no power to retain its contents.

* NVRAM (Non-Volatile Radom Access Memory), type of electronic disk.

It is same as DRAM with battery backup power. This memory can be as fast as DRAM but has a limited duration in which it is non-volatile.

* Memory system must use only as much expensive memory as necessary while providing as much inexpensive, non-volatile memory as possible.
* Caches can be installed to improve performances where a large access-time or transfer-rate disparity exists between two components.

1. INPUT/ OUTPUT STRUCTURE

* Storage is only one of many input/ output devices within a computer. A large portion of O.S. code is dedicated to managing input output, both because of importance to reliability and performance of system and because of varying nature of devices.
* A general purpose computer system consists of CPUS’s and multiple device controllers that are connected through a common bus. Each device controller is in charge of specific type of device. Depending on the controller, there may be more than one attached device. For instance, seven or more devices can be attached to small computer systems interface (SCSI)
* Device Controller: it is responsible for moving data between the peripheral devices that it controls and its local buffer storage.
* Device Driver: loads the appropriate registers within the device controller.
* The device controller, in term, examines the contents of these registers to determine what actions to take. Controller starts the transfer of data from device to buffer. Once the transfer of data is complete, the device controller informs the device drier via an interrupt that it has finished its operation.

Device driver then returns the control to O.S. for other operations, device driver returns status information.

* This form of interrupt driver I/O is fine for moving small amounts of data but can produce high overhead for bulk data movement. To solve this problem data memory access (DMA) is used. After setting up buffers and counters for 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 is generated, per block, to tell device driver that the operation is completed, rather than one interrupt per byte.