Input/ Output

UNIT-5

Introduction:

- The I/O subsystem of a computer provides an efficient mode of communication between the central system and the outside environment.
- It handles all the input-output operations of the computer system.

Peripheral Devices

- Input or output devices that are connected to computer are called peripheral devices.
- These devices are designed to read information into or out of the memory unit upon command from the CPU and are considered to be the part of computer system. These devices are also called peripherals.

For example: Keyboards, display units and printers are common peripheral devices.

There are three types of peripherals:

1. **Input peripherals** : Allows user input, from the outside world to the computer.

   Example: Keyboard, Mouse etc.

2. **Output peripherals**: Allows information output, from the computer to the outside world.

   Example: Printer, Monitor etc

3. **Input-Output peripherals:** Allows both input(from outised world to computer) as well as, output(from computer to the outside world).

   Example: Touch screen etc

What is an interrupt?

- An interrupt is a signal for the CPU to stop what it is currently doing and do something else as a higher priority
- The CPU is in a continuous loop of carrying out the fetch-decode-execute cycle, however there are occasions when this needs to be interrupted

Types of Interrupt:

- An interrupt can be generated by hardware and software:
    i. Hardware Interrupts
    ii. Software Interrupts
    iii. Timer Interrupts
    iv. I/O Interrupts
- Interrupts are added to an area called the interrupt service routine
- The interrupt service routine holds instructions that will need to be fetched, decoded and executed to complete the commands of the interrupt
- The contents of the registers within the CPU cannot be lost by an interrupt, so contents are copied to a reserved area in RAM called a stack
- Contents are added to the top of the stack, which will save them for later retrieval when the interrupt is complete
- The interrupt will be executed instead of the original instructions

1. Hardware Interrupts

- These interrupts are triggered by external hardware devices to signal the processor about a specific event.
- Generated when a hardware device (e.g., keyboard, mouse, printer) requires the CPU's attention.
- The device sends a signal to the processor via the Interrupt Request (IRQ) line.
- The CPU temporarily halts its current task and jumps to the Interrupt Service Routine (ISR) to handle the request.

Examples:

- Keyboard Interrupt: Triggered when a key is pressed.
- Mouse Interrupt: Triggered by mouse movements or clicks.
- Printer Interrupt: Triggered when a printer signals that it has completed printing a page or is out of paper.

2. Software Interrupts

- These interrupts are initiated by software instructions to request specific services from the operating system.
- A program executes an interrupt instruction (INT) to call the operating system or interrupt handler.
- Commonly used to switch between user and kernel modes or request OS services.
- Software Interrupts occurs as part of the program's execution flow and Typically lower priority compared to hardware interrupts.

Examples:

- System Calls: A program requesting file access or memory allocation.
- Error Interrupts: Triggered by errors such as divide-by-zero or invalid memory access.

3. Timer Interrupts

- These interrupts are triggered by an internal timer within the CPU or system.
- A programmable timer generates interrupts at regular intervals.
- The CPU responds by executing the timer interrupt handler.

Examples:

- Task Scheduling: Operating systems like Linux use timer interrupts for multitasking by switching between processes at fixed intervals.
- Periodic Tasks: Triggering periodic updates, such as refreshing a display or saving data automatically.

4. I/O Interrupts

- These interrupts occur when an I/O operation completes or requires CPU intervention.
- When an I/O device (e.g., disk, network card) finishes a task or encounters an issue, it sends an interrupt signal.
- The CPU handles this interrupt by running the I/O interrupt handler to either process the data or address the issue.

Examples:

- Disk Interrupts: Triggered when data is read from or written to the disk.
- Network Interrupts: Triggered when a network packet is received or transmitted.
- USB Interrupts: Triggered when a USB device connects or disconnects.

Modes of I/O Data Transfer

- Data transfer between the central unit and I/O devices can be handled in generally three types of modes which are given below:
    1. Programmed I/O
    2. Interrupt Initiated I/O
    3. Direct Memory Access

Programmed I/O

- Programmed I/O instructions are the result of I/O instructions written in computer program.
- Each data item transfer is initiated by the instruction in the program.
- Usually the program controls data transfer to and from CPU and peripheral.
- Transferring data under programmed I/O requires constant monitoring of the peripherals by the CPU.

Interrupt Initiated I/O

- In the programmed I/O method the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer.
- This is time consuming process because it keeps the processor busy needlessly.
- This problem can be overcome by using interrupt initiated I/O.
- In this when the interface determines that the peripheral is ready for data transfer, it generates an interrupt.
- After receiving the interrupt signal, the CPU stops the task which it is processing and service the I/O transfer and then returns back to its previous processing task.

Direct Memory Access (DMA)

- Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This technique is known as DMA.
- In this, the interface transfer data to and from the memory through memory bus.
- A DMA controller manages to transfer data between peripherals and memory unit.
- Many hardware systems use DMA such as disk drive controllers, graphic cards, network cards and sound cards etc. It is also used for intra chip data transfer in multicore processors.
- In DMA, CPU would initiate the transfer, do other operations while the transfer is in progress and receive an interrupt from the DMA controller when the transfer has been completed