CS_252 Computer Organization and Assembly Language

Course Instructor: Sunbal Rani

Week No 5

Contents

Introduction to I/O Systems

An I/O system is a combination of hardware and software that allows data to move into (input) and out of (output) a computer system.

Bridging the Gap: CPU operates much faster than I/O devices.

Device Diversity: Systems handle devices with different speeds, formats, and communication methods.

Examples:

Fast devices: Hard drives, network cards

Slow devices: Keyboard, mouse

Overview of I/O Models

I/O models define how data moves between CPU and I/O devices.

Goals: Balance performance, resource use, and responsiveness.

Common I/O Models:

- Programmed I/O (Polling)
- Interrupt-Driven I/O
- Direct Memory Access (DMA)
- Memory-Mapped I/O
- Channel I/O

Programmed I/O (Polling)

Mechanism:

- CPU continuously checks device status (polls).
- Data transfer occurs when the device is ready.

Steps:

- CPU sends a command to the I/O device.
- CPU waits in a loop until the device is ready.
- Data is transferred between device and memory

Programmed I/O (Polling) – Pros & Cons

Advantages:

- Simple to implement.
- CPU has full control over I/O.

Disadvantages:

- Inefficient CPU utilization due to busy waiting.
- Best suited for systems with fast or low-priority I/O operations.

Interrupt-Driven I/O

Mechanism:

- CPU issues command and continues other tasks.
- Device sends interrupt when ready.

Steps:

- CPU sends I/O command.
- Device interrupts CPU when ready.
- CPU pauses, performs I/O, resumes work.

Interrupt-Driven I/O – Pros & Cons

Advantages:

- Improved CPU utilization.
- Efficient for slower I/O devices.

Disadvantages:

- Interrupt overhead (save/restore CPU state).
- Complex interrupt handling for multiple devices.

Direct Memory Access (DMA)

Mechanism:

- DMA controller handles data transfer directly between memory and device.
- CPU only initiates and finalizes transfer.

Steps:

- CPU sets up DMA with source, destination, and size.
- DMA controller performs transfer.
- DMA signals CPU upon completion.

DMA – Pros & Cons

Advantages:

- Minimal CPU involvement.
- Enables high-speed data transfer.

Disadvantages:

- Requires extra DMA hardware.
- Complex coordination among multiple devices.

Memory-Mapped I/O

Mechanism:

- Device registers treated as regular memory locations.
- CPU uses normal memory instructions for I/O.

Steps:

- I/O device assigned specific memory addresses.
- CPU reads/writes to these addresses for communication.

Memory-Mapped I/O – Pros & Cons

Advantages:

- Simple CPU-device interaction; no special I/O instructions.
- Typically, faster due to memory-based access.

Disadvantages:

- Reduces available memory space.
- Complex memory and I/O management.

Channel I/O

Mechanism:

- Uses dedicated processors (I/O channels) for device communication.

Steps:

- CPU issues high-level I/O command to channel.
- I/O channel manages data transfer.
- Channel signals CPU when task completes.

Channel I/O – Pros & Cons

Advantages:

- Very efficient; frees CPU for other tasks.
- Ideal for large and complex systems.

Disadvantages:

- High cost and complexity.
- Primarily used in mainframes and servers.

Comparison of I/O Models

I/O Model	CPU Involvement	Efficiency	Use Case
Programmed I/O	High	Low (wasted CPU cycles)	Simple systems, fast devices
Interrupt-Driven I/O	Medium	Higher than polling	Slower devices, responsive systems
Direct Memory Access	Low	Very high	High-speed devices (disk, network)
Memory-Mapped I/O	Low	High	Systems with limited I/O instruction set
Channel I/O	Minimal	Extremely high	Large mainframes, high I/O environments

Thank You