|  |
| --- |
| Experiment No. 9 |
| Implement a program to demonstrate various data transfer techniques |
| Date of Performance: |
| Date of Correction: |

**Aim:** To implement a program that demonstrates various data transfer techniques used in computer systems for moving data between memory, registers, and I/O devices.

**Objective:** To understand and implement different data transfer methods used in computer architecture such as programmed I/O, interrupt-driven I/O, and direct memory access (DMA).

**Theory:**

**Introduction:**

In computer systems, data transfer refers to the movement of data between different components such as:

* CPU ↔ Memory
* CPU ↔ I/O devices
* Memory ↔ I/O devices

Efficient data transfer is essential for system performance, especially when interacting with high-speed or real-time devices. There are three primary data transfer techniques used in computer architecture:

**1. Programmed I/O (Polling Method):**

**In this method:**

* The CPU is in control of all I/O operations.
* The CPU continuously polls (checks) the status of I/O devices.
* When the device is ready, the CPU transfers data between the I/O port and registers/memory.

**Pros:**

* Simple to implement.

**Cons:**

* CPU remains busy in checking device status.
* Inefficient for high-speed or multiple I/O devices.

**Example in Code:**The program reads or writes data by directly accessing I/O ports or memory addresses.

**2. Interrupt-Driven I/O:**

**In this method:**

* The CPU does not poll the I/O devices continuously.
* I/O devices send an interrupt signal to the CPU when they are ready.
* The CPU then pauses current execution and handles the interrupt via an Interrupt Service Routine (ISR).

**Pros:**

* CPU time is utilized more efficiently.
* More responsive for real-time systems.

**Cons:**

* Slightly more complex to implement than programmed I/O.

**Real-life analogy:** A doorbell. Instead of checking the door repeatedly, you get notified when someone is at the door.

**3. Direct Memory Access (DMA):**

**In this method:**

* A separate DMA controller is used to transfer data directly between memory and I/O devices without CPU involvement.
* The CPU only initiates the transfer and gets notified once it is complete.

**Pros:**

* Efficient for bulk data transfer.
* Frees up the CPU to do other tasks during data transfer.

**Cons:**

* Requires additional hardware (DMA controller).

**Use cases:** Transferring large data blocks like disk read/write operations, audio/video streaming.

**Importance in Computer Architecture:**

* These techniques allow efficient utilization of the processor and I/O devices.
* They play a crucial role in system performance, especially in embedded systems, real-time applications, OS kernel development, and device drivers.
* Understanding these methods is essential for designing interrupt controllers, I/O-mapped hardware, and low-level software.

**Solution:**

**Conclusion:** We explored and simulated different data transfer methods such as programmed I/O, interrupts, and DMA, and understood their roles in efficient communication between components.