Name:- Tazmeen Afroz Roll NO:- 22P-9252

SECTION:- BAI-5A

COURSE:- Operating Systems

ASSIGNMENT NO 1

1. Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.

a) How does the CPU interface with the device to coordinate the transfer?

The CPU communicates with the DMA (Direct Memory Access) controller by configuring its registers with details such as the source and destination addresses, the size of the data to be transferred, and any necessary control information. Once set up, the DMA controller autonomously manages the data transfer between the memory and the I/O device, eliminating the need for further CPU involvement during the transfer.

b) How does the CPU know when the memory operations are complete?

After completing the data transfer, the DMA controller sends an interrupt signal to the CPU. This interrupt alerts the CPU that the memory operations have finished, allowing it to proceed with any subsequent tasks that depend on the completion of the data transfer.

c) Does this process interfere with the execution of user programs? If so, describe what forms of interference are caused.

Yes, while the DMA controller is performing the data transfer, it may temporarily take control of the memory bus, which can prevent the CPU from accessing memory during those brief periods. This can cause minor delays or a slight decrease in performance for user programs that require memory access at the same time as the DMA operation.

Q2) Difference between symmetric and asymmetric multiprocessing. What are the advantages and disadvantages of multiprocessor systems?

	Symmetric Multiprocessing	Asymmetric Multiprocessing
1 **Processor Roles.**	All processors are equal peers; any processor can perform any task.	Processors hava specific roles; the master controls the system, and slaves perform assigned tasks.
2 **Memory Access.**	Shared memory model with uniform access times.	May have separate memory spaces access times can vary.
3 **Operating System**	A single OS instance manages all processors.	The master processor runs the OS, slaves may run minimal or specialized code.
4 **Task Scheduling**	The OS schedular distributes tasks evenly among processors.	The master processor assigns tasks to slaves as needed
5 **Scalability**	Limited by the potential for resource contention as more processors share the same bus and memory.	Can be more scalable by isolating tasks and resources, reducing contention.

(b)

Advantages

1 **Enhanced Performance:**
Multiple processors allow for parallel execution of tasks, which speeds up processing and increases system throughout.

2 **Improved Reliability:**
If one processor fails, other can continue to function, providing fault tolerance and reducing system downtime.

3 **Scalability:**
Additional processors can be added to the system to handle increased workloads, making it easier to scale performance as needed.

Disadvantages

Increased Complexity:
Designing and programming for multiprocessor systems is more complex due to the need for effective synchronization and coordination between processors.

Resource Contention.
Processors may compete for access to shared resources like memory and I/O, which can lead to bottlenecks and reduced performance.

Higher Costs:
The additional hardware components and the complexity involved in maintaining multiprocessor systems leads to increased overall costs.

Q3) What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?

Interrupts serve to notify the CPU of immediate events or conditions that need attention, allowing the system to handle asynchronous hardware or software events efficiently without the need for constant polling by the CPU.

	Interrupt	Trap
1 - Origin	Typically generated by hardware devices and occus asynchronously, independent of the program's current execution.	Generated by software as a result of executing certain instructions, making them synchronous with the program flow.
2 - Function	Handle external hardware events, such as input from peripherals or hardware timers.	Manage exceptions like errors (e.g division by zero) or facilitate system calls to request operating system services.
3. Handling Mechanism	Invoke interrupt service routines or handlers designed to respond to hardware signals.	Trigger trap or exception handlers within the operating system to process exceptions or execute system calls.

Generated by User Programs.

Yes, user programs can intentionally generate traps to perform system calls. This mechanism allows programs to safely request services from the operating system, enabling them to execute ~~pre~~ privileged operations under controlled conditions.

Q4) Rank the following storage system from slowest to fastest.

Magnetic Tapes (Slowest)
Optical Disks
Hard Disk Drives
Nonvolatile Memory
Main Memory
Cache
Registors (Fastest)

Q5) Difference between multiprogramming systems and multiprocessing systems.

Multiprogramming:

Involves loading several programs into memory and having one processor manage their execution by time-sharing, effectively handling multiple tasks sequentially.

Multiprocessing:

Involves multiple processors working concurrently to execute multiple tasks at the same time, providing true parallelism and increased computational speed.

Q6) Give two reasons why caches are useful ... device?

Enhanced Performance:

By storing critical data locally, caches allow for quicker data access, which boosts the overall speed of applications and system responsiveness.

Reduced Memory Bottlenecks:

Caches alleviate the load on main memory or storage devices by handling frequent data requests, thus improving system throughput.

Problems they solve:-

Latency Reduction:

They decrease the delay caused by slower storage devices, ensuring the CPU spends less time waiting for data.

Efficient Resource Utilization:

Caches optimize the use of system resources by ~~proti~~ prioritizing fast access to commonly used data.

Problems they cause

Cache Misses:

If the required data isn't in tho cache, it can lead to delays while fetching data from slower memory, potentially impacting performance.

Synchronization Overhead:

Maintaing consistency between the cache and the main memory introduces additional overhead and complexity in system design.

Why not make the cache as large as the device.

Economic Constraints

The high cost of cache memory technologies makes it financially unfeasible to build caches as large as main memory or storage devices.

Design Trades-Offs :

Caches are optimized for speed rather than capacity, while main memory and storage devices are designed to provide large capacity at lower costs

Q7) Distinguish betwoen the client-server and peer-to-peer models of distributed systems.

Client - Server Model:

Characterized by a central server that delivers services to multiple client machines, it is suitable for both small and large networks but can be more expensive due to the need for dedicated server hardware. Clients and servers have distinct roles, with the server handling data storage and management.

Peer-to-Peer Model:

Features a network where all nodes are equal and can request and provide services simultaneously. It's typically more affordable and easier to set up but is generally used for smaller networks due to potential o stability ~~issues~~ issues without centralized management.

Q8) What is the purpose of system call, and how ..... operation?

Purpose:

They act as a bridge between user applications and the operating system, allowing programs to access low-level hardware and kernal services necessary for tasks like reading files, creating processes, or communicating over networks

Relation to OS and Dual-Mode Operation:

System calls are essential to dual-mode operation by triggering a switch from user mode to kernal mode. This mechanism ensures that user programs can request kernal services without direcly accessing or compromising the protected kernal space, maintaing system stability and security