Operating Systems-1: CS3510

Name: Saksham

Roll No.: AI22BTECH11024

Theory Assignment 1: Chapter 1

Question 1:

a) What is the purpose of interrupts?

Ans) An interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention. It alerts the processor to a high-priority process requiring interruption of the current working process. In I/O devices, one of the bus control lines is dedicated for this purpose and is called the Interrupt Service Routine (ISR).

b) How does an interrupt differ from a trap?

Ans) The primary difference between an interrupt and a trap lies in their triggers and purposes. A trap is a synchronized software-generated interrupt caused either by an error (for example, division by zero or invalid memory access) or by a specific request from a user program that an operating-system service be performed by executing a special operation called a system call. In contrast, an interrupt is an asynchronous signal generated by hardware devices to demand immediate attention from the CPU, initiating hardware interrupt routines. While traps are generated by user programme instruction, interrupts are external events triggered by hardware, serving distinct roles in the interaction between software and hardware within a computing system.

c) Can traps be generated intentionally by a user program?

Ans) Yes, traps can be generated intentionally by a user program.

d) If so, for what purpose?

Ans) A user program may purposely create traps in order to carry out debugging operations, which involve using traps to find and fix mathematical mistakes or other problems in the code. Furthermore, when a user program

wants to run code in kernel mode or ask the operating system for a specific function, it can also produce traps by issuing system calls. Therefore, traps act as a purposeful means by which user programs can communicate with the operating system, allowing for regulated communication and the performance of specific tasks.

Question 2:

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?

Ans) Through the use of DMA, the CPU can communicate with an I/O device by first providing the DMA controller with transfer details, after which the device will initiate a DMA request, the DMA controller will mediate if necessary, seize control of the system bus, oversee data transfer between the device and memory directly, generate an interrupt if necessary, and then return control of the system bus to the CPU. This lessens the execution burden on the CPU by enabling it to transfer data transfer duties to the DMA controller.

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

Ans) The CPU knows when memory operations are complete in DMA through interrupts, where the DMA controller signals the CPU upon completion.

c) The CPU is allowed to execute other programs while the DMA controller is transferring data. Does this process interfere with the execution of the user programs? If so, describe what forms of interference are caused.

Ans) The DMA controller and the CPU are both bus masters. If the DMA controller and the CPU wanted to access the memory at the same time, that would be an issue. Accordingly, when the DMA controller seizes the memory bus, the CPU should be temporarily blocked from accessing main memory. However, if the CPU and the DMA controller update the same memory locations and the CPU is still permitted to access data in its primary and secondary caches, a coherency problem could arise.
