9 Lecture - CS401

Important Subjective

1. **What is a hardware interrupt and how is it different from a software interrupt?**
   Answer: A hardware interrupt is a signal generated by a device to request attention from the CPU. It is triggered by a hardware event such as an I/O request or a timer expiration. A software interrupt, on the other hand, is generated by a software instruction and is used for system calls or to handle exceptions.

   How are hardware interrupts prioritized and handled by the CPU?
   Answer: Hardware interrupts are prioritized by the interrupt controller and are handled by the operating system. When a hardware interrupt occurs, the CPU stops executing its current program and transfers control to the corresponding interrupt handler.

   What is a non-maskable interrupt and why is it important?
   Answer: A non-maskable interrupt is a type of hardware interrupt that cannot be disabled by software. It is used for critical events that cannot be ignored, such as power failures or hardware errors. Non-maskable interrupts are important because they ensure that the system can respond to critical events in a timely manner.

   How does the interrupt controller manage and prioritize hardware interrupts?
   Answer: The interrupt controller is responsible for managing and prioritizing hardware interrupts. It receives signals from devices and assigns priority levels to each interrupt. When multiple interrupts occur simultaneously, the interrupt controller determines which one should be handled first based on its priority level.

   What is interrupt latency and how can it be minimized?
   Answer: Interrupt latency is the time between the occurrence of a hardware interrupt and the start of its corresponding interrupt handler. It can be minimized by using techniques such as interrupt preemption, which allows a higher-priority interrupt to interrupt a lower-priority interrupt, and interrupt chaining, which allows multiple interrupts of the same type to be handled in sequence.

   How are interrupts handled in a multi-core processor?
   Answer: In a multi-core processor, each core has its own interrupt controller and can handle interrupts independently. The operating system must coordinate the handling of interrupts across all cores to ensure that they are handled in a timely and efficient manner.

   What is the difference between a vectored interrupt and a non-vectored interrupt?
   Answer: A vectored interrupt is an interrupt that provides information about the source of the interrupt to the CPU. This information is used to determine the corresponding interrupt handler. In contrast, a non-vectored interrupt does not provide this information and requires the CPU to search for the appropriate interrupt handler.

   How does interrupt masking work and why is it used?
   Answer: Interrupt masking is a technique used to disable or block interrupts of a certain type or

with a certain priority level. It is used to prevent interrupts from interfering with critical tasks or to ensure that certain interrupts are handled before others.

What is interrupt chaining and how is it used to handle multiple interrupts of the same type?
Answer: Interrupt chaining is a technique used to handle multiple interrupts of the same type. When multiple interrupts of the same type occur, the interrupt handler for the first interrupt is executed, and then the handler for the second interrupt is called from within the first handler. This process is repeated until all interrupts have been handled.

How do device drivers interact with interrupts and what role do they play in interrupt handling?
Answer: Device drivers are responsible for managing the interaction between hardware devices and the operating system. They register interrupt handlers for their associated devices and are responsible for handling interrupts generated by those devices. Device drivers play a critical role in interrupt handling as they ensure that the system can respond to hardware events in a timely and efficient manner.