Homework 10

11/28/23

AIDAN KISER

1. Interrupts are signals sent to the processor by external devices or internal processes to request its attention. They are crucial for handling real-time events, improving system efficiency, and enabling multitasking

- Hardware:
  - Hardware interrupts are signals generated by external hardware devices to request attention from the CPU.
  - Examples include timer, Input/Output, and hardware fault interrupts.
- Software:
  - Software interrupts are signals that are generated by executing specific software instructions. They are typically used for system calls and to invoke other operating system functions.
  - Examples include system calls, breakpoint, and software exception interrupts.
- Maskable:
  - Maskable interrupts can be temporarily ignored or disabled by the CPU.
  - Examples include external hardware and peripheral device interrupts.
- Non-Maskable:
  - Non-maskable interrupts cannot be ignored or disabled by the CPU. They always demand immediate attention.

- Examples include hardware malfunctions and watchdog timer interrupts.

2. Interrupt vectoring is a mechanism used to handle the interrupts efficiently. It helps the system determine the appropriate routine or handler to execute in response to the interrupt. Here is the step-by-step process:

1) Interrupt occurs:

• An external device or internal process generates an interrupt signal, indicating that it requires attention from the CPU.

2) Interrupt Request

• The IRQ is sent to the CPU.

3) Interrupt Controller

• Prioritizes interrupts and presents the highest priority interrupt to the CPU.

4) Interrupt Vector

• Each interrupt is associated with a unique interrupt vector. A vector is a data structure or table that contains information about the interrupt and the address of the corresponding interrupt handler or routine.

5) Interrupt Vector Table

• The IVT is a specific region of memory that holds the interrupt vectors.

6) Interrupt Vector Number:

- The CPU reads the interrupt vector number from the interrupt controller.

7) Indexing the Interrupt Vector Table:

- The CPU uses the interrupt vector number as an index to access the corresponding entry in the IVT.

8) Interrupt Handler Execution:

- The entry in the IVT contains the address of the interrupt handler or routine associated with the specific interrupt. The CPU jumps to this address, initiating the execution of the interrupt handler.

9) Interrupt Handling:

- The interrupt handler performs the necessary tasks to service the interrupt.

10) Return from Interrupt:

- Once the interrupt handler completes its tasks, it executes a return-from-interrupt instruction. The instruction restores the saved state of the interrupted process and allows it to resume normally.

11) Resuming Normal Execution:

- After handling the interrupt, the CPU resumes execution where it was interrupted.