Interrupts

The 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)*.
When a device raises an interrupt at let's say process i, the processor first completes the execution of instruction i. Then it loads the Program Counter (PC) with the address of the first instruction of the ISR. Before loading the Program Counter with the address, the address of the interrupted instruction is moved to a temporary location. Therefore, after handling the interrupt the processor can continue with process i+1.

While the processor is handling the interrupts, it must inform the device that its request has been recognized so that it stops sending the interrupt request signal. Also, saving the registers so that the interrupted process can be restored in the future, increases the delay between the time an interrupt is received and the start of the execution of the ISR. This is called Interrupt Latency.

Hardware Interrupts:

In a hardware interrupt, all the devices are connected to the Interrupt Request Line. A single request line is used for all the n devices. To request an interrupt, a device closes its associated switch. When a device requests an interrupt, the value of INTR is the logical OR of the requests from individual devices.

The sequence of events involved in handling an IRQ:

1. Devices raise an IRQ.
2. The processor interrupts the program currently being executed.
3. The device is informed that its request has been recognized and the device deactivates the request signal.
4. The requested action is performed.
5. An interrupt is enabled and the interrupted program is resumed.

Handling Multiple Devices:

When more than one device raises an interrupt request signal, then additional information is needed to decide which device to be considered first. The following methods are used to decide which device to select: Polling, Vectored Interrupts, and Interrupt Nesting. These are explained as following below.

1. **Polling:** In polling, the first device encountered with the IRQ bit set is the device that is to be serviced first. Appropriate ISR is called to service the same. It is easy to implement but a lot of time is wasted by interrogating the IRQ bit of all devices.
2. **Vectored Interrupts:** In vectored interrupts, a device requesting an interrupt identifies itself directly by sending a special code to the processor over the bus. This enables the processor to identify the device that generated the interrupt. The special code can be the starting address of the ISR or where the ISR is located in memory and is called the interrupt vector.

3. **Interrupt Nesting:** In this method, the I/O device is organized in a priority structure. Therefore, an interrupt request from a higher priority device is recognized whereas a request from a lower priority device is not. The processor accepts interrupts only from devices/processes having priority.

Processors’ priority is encoded in a few bits of PS (Process Status register). It can be changed by program instructions that write into the PS. The processor is in supervised mode only while executing OS routines. It switches to user mode before executing application programs.