

1.6 Interrupt Handling:

Interrupts inform the processor that an external device or an associated task requires immediate attention of the CPU.

- ❖ Interrupts can be either Synchronous or Asynchronous.
- ❖ Interrupts which occurs in sync with the currently executing task is known as Synchronous interrupts. Usually the software interrupts fall under the Synchronous Interrupt category. Divide by zero, memory segmentation error etc are examples of Synchronous interrupts.
- ❖ For synchronous interrupts, the interrupt handler runs in the same context of the interrupting task.
- ❖ Asynchronous interrupts are interrupts, which occurs at any point of execution of any task, and are not in sync with the currently executing task.
- ❖ The interrupts generated by external devices (by asserting the Interrupt line of the processor/controller to which the interrupt line of the device is connected) connected to the processor/controller, timer overflow interrupts, serial data reception/ transmission interrupts etc are examples for asynchronous interrupts.
- ❖ For asynchronous interrupts, the interrupt handler is usually written as separate task (Depends on OS Kernel implementation) and it runs in a different context. Hence, a context switch happens while handling the asynchronous interrupts.
- ❖ Priority levels can be assigned to the interrupts and each interrupts can be enabled or disabled individually.
- ❖ Most of the RTOS kernel implements ‘Nested Interrupts’ architecture. Interrupt nesting allows the pre-emption (interruption) of an Interrupt Service Routine (ISR), servicing an interrupt, by a higher priority interrupt.