



SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

COIMBATORE-35

Accredited by NBA-AICTE and Accredited by NAAC – UGC with A++ Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



19EEE305 / EMBEDDED SYSTEMS III YEAR / VI SEMESTER

UNIT-IV: RTOS BASED EMBEDDED SYSTEM DESIGN

TASK SCHEDULER, MANAGEMENT, INTERRUPT





TASK SCHEDULER



Task scheduler is one of the important component of the Kernel. The task scheduler establishes task time slots .

Every RTOS provides three specific functions.

- (i).Scheduling (ii) Dispatching and (iii). Inter-process communication and synchronization.

Scheduling Algorithms: In Multitasking system to schedule the various tasks , different scheduling algorithms are used..They are

- (a).First in First out
- (b).Round Robin algorithm
- (c).Round Robin with priority
- (d)Non-preemptive
- (e)Pre-emptive.



TASK SCHEDULER



FCFS:

- FCFS also known as first in first out (FIFO) is the simplest scheduling policy. Arriving jobs are inserted into the tail (rear) of the ready queue and process to be executed next is removed from the head (front) of the queue.

NON – PRE EMPTIVE SCHEDULING:

- Non- Preemptive scheduling is employed in non-preemptive multitasking systems. In this scheduling type, the currently executing task/process is allowed to run until it terminates or enters the wait state waiting for an I/O or system resource.

PRE EMPTIVE SCHEDULING:

- In preemptive mode, currently running process may be interrupted and forces the currently active process to release the CPU on certain events such as a clock interrupt, some I/O interrupts or a system call and they moved to the ready state by the OS.

TASK MANAGEMENT



Task Management is the component of the Kernel.

- Create a task
- Delete a task
- Suspend a task
- Resume a task
- Change priority of a task
- Query a task
- Interrupts is a mechanism for alleviating the delay caused by uncertainty and for maximizing system performance.





INTERRUPT

Interrupts is a mechanism for alleviating the delay caused by uncertainty and for maximizing system performance.

- Instead of polling the device or entering a wait state, the CPU continuously executing its instruction and performing useful work

Interrupt Service Routine (ISR) have the higher priorities over the RTOS function and the tasks. So ISR should not wait for a semaphore, mailbox, message or queue message.

- An ISR should not also wait for mutex, else it has to wait for other critical section code to finish before the critical codes in the ISR can run.

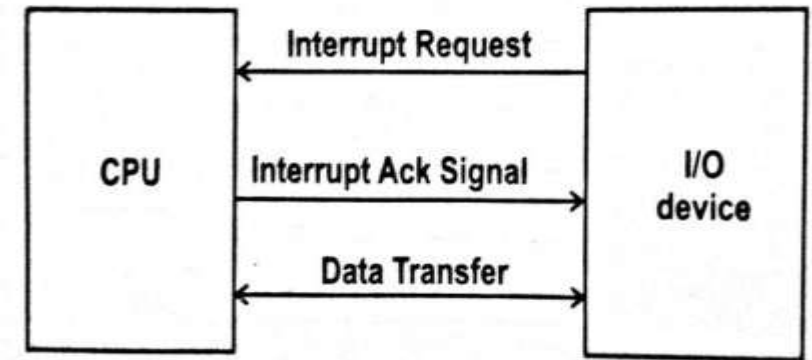
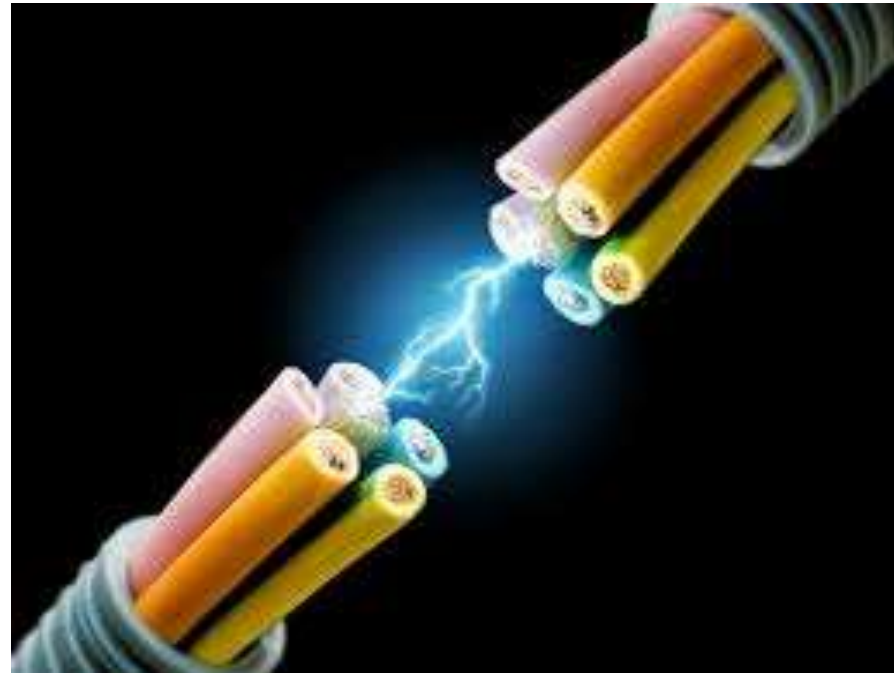


Fig. Handshaking signals of Interrupts





RECAP....



...THANK YOU

