REAL TIME OPERATING SYSTEM

TOPICS OF DISCUSSION.

- WHAT IS RTOS.
- COMPARISON BETWEEN RTOS AND GENERAL OPERATING SYSTEMS.
- TYPES OF RTOS.
- CHARACTERISTICS OF RTOS.
- FUNCTIONS OF RTOS.
- APPLICATIONS OF RTOS.
- EXAMPLE OF SOME RTOS
- CONCLUSION.

What is Real Time ?

- " Real time in operating systems:

  The ability of the operating system to provide a required level of service in a bounded response time."

  - POSIX Standard 1003.1

WHAT IS RTOS.

- It responds to inputs immediately(Real-Time).
- Here the task is completed within a specified time delay.
- In real life situations like controlling traffic signal or a nuclear reactor or an aircraft,
- The operating system has to respond quickly.

What a RTOS is not

- Real time computing is equivalent to fast computing.
- Real time systems operate in a static environment
- Real time programming involves assembly coding, priority interrupt programming, writing device drivers.

Soft RTOS…

- In a soft real-time system, it is considered undesirable, but not catastrophic, if deadlines are occasionally missed.
- Also known as “best effort” systems
- Most modern operating systems can serve as the base for a soft real time systems.
- Examples:
  - multimedia transmission and reception,
  - networking, telecom (cellular) networks,
  - web sites and services
  - computer games.

Hard RTOS…

- A hard real-time system has time-critical deadlines that must be met; otherwise a catastrophic system failure can occur.
- Absolutely, positively, first time every time
- Requires formal verification/guarantees of being to always meet its hard deadlines (except for fatal errors).
- Examples:
  - air traffic control
  - vehicle subsystems control
  - Nuclear power plant control

CHARACTERISTICS OF RTOS.

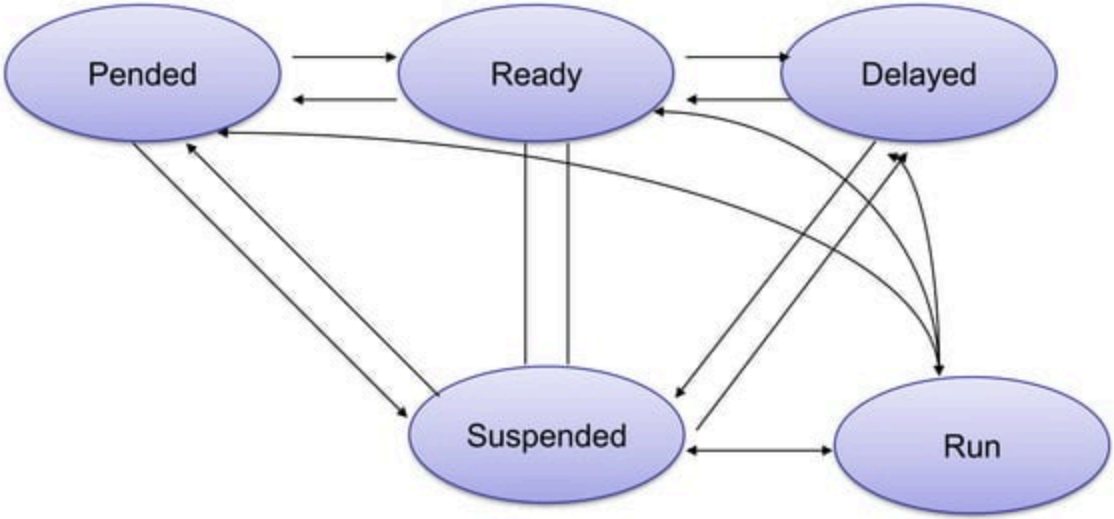
FUNCTIONS OF RTOS

- Task management
- Scheduling.
- Resource Allocation.
- Interrupt Handling.

Task management

- In Real Time Applications the Process is called as Task which takes execution time and occupies memory.
- Task management is the process of managing tasks through its life cycle.

Task States



Task/Process States

- Each task/Process can belong to one and only one state
- The Scheduler only operates on the processes in the Ready state
- There is a single process in the Run/current state at any time.
- Transitions to and from the Ready queue are affected as a part of the execution of the RTOS resource/object services or as a result of timing events

Typical Task Operations

- creating and deleting tasks,
- controlling task scheduling, and
- obtaining task information.

Scheduling in RTOS

- More information about the tasks are known
  - No of tasks
  - Resource Requirements
  - Release Time
  - Execution time
  - Deadlines
- Being a more deterministic system better scheduling algorithms can be devised.

Scheduling Algorithms in RTOS

- Clock Driven Scheduling
- Weighted Round Robin Scheduling
- Priority Scheduling
  (Greedy / List / Event Driven)

Scheduling Algorithms in RTOS (contd)

- Clock Driven
  - All parameters about jobs (release time/ execution time/deadline) known in advance.
  - Schedule can be computed offline or at some regular time instances.
  - Minimal runtime overhead.
  - Not suitable for many applications.

Scheduling Algorithms in RTOS (*contd*)

- Weighted Round Robin
  - Jobs scheduled in FIFO manner
  - Time quantum given to jobs is proportional to it's weight
  - Example use : High speed switching network
    - QOS guarantee.
  - Not suitable for precedence constrained jobs.
    - Job A can run only after Job B. No point in giving time quantum to Job B before Job A.

Scheduling Algorithms in RTOS (*contd*)

- Priority Scheduling
  (Greedy/List/Event Driven)
  - Processor never left idle when there are ready tasks
  - Processor allocated to processes according to priorities
  - Priorities
    - static - at design time
    - Dynamic - at runtime

Priority Scheduling

- Earliest Deadline First (EDF)
  - Process with earliest deadline given highest priority
- Least Slack Time First (LSF)
  - slack = relative deadline – execution left
- Rate Monotonic Scheduling (RMS)
  - For periodic tasks
  - Tasks priority inversely proportional to it's period

Resource Allocation in RTOS

- Resource Allocation
  - The issues with scheduling applicable here.
  - Resources can be allocated in
    - Weighted Round Robin
    - Priority Based
- Some resources are non preemptible
  - Example : semaphores
- Priority Inversion if priority scheduling is used

Other RTOS issues

- Interrupt Latency should be very small
  - Kernel has to respond to real time events
  - Interrupts should be disabled for minimum possible time
- For embedded applications Kernel Size should be small
  - Should fit in ROM
- Sophisticated features can be removed
  - No Virtual Memory
  - No Protection

INTERRUPTS HANDLING OF RTOS.

- An interrupt is a signal from a device attached to a computer or from a program with in a computer that causes the main program that is operating system to stop and figure out what to do next.

- Interrupts cause the processor to suspend the operations whatever it is doing instead execute the code that will respond to the event whatever caused the interrupt.

APPLICATIONS OF RTOS.

- Almost all the modern telecommunication systems make use of RTOS .
- Radar systems, network switching control systems, satellite monitoring systems, satellite launch-control and maneuvering mechanisms, global positioning systems all have their roots in RTOS.
- Now a days RTOS are increasingly finding use in strategic and military operations. These are used in guided missile launching units, track-and-trace spy satellites, etc.