Real Time Operating systems (RTOS) concepts

Abu Bakr Mohamed Ramadan
Eng.abubakr88@gmail.com

Content:

- RTOS Kernel,

- Priority Based Kernel

- RTOS Task Status,

- Preemptive Kernel,

- Non-Preemptive Kernel,
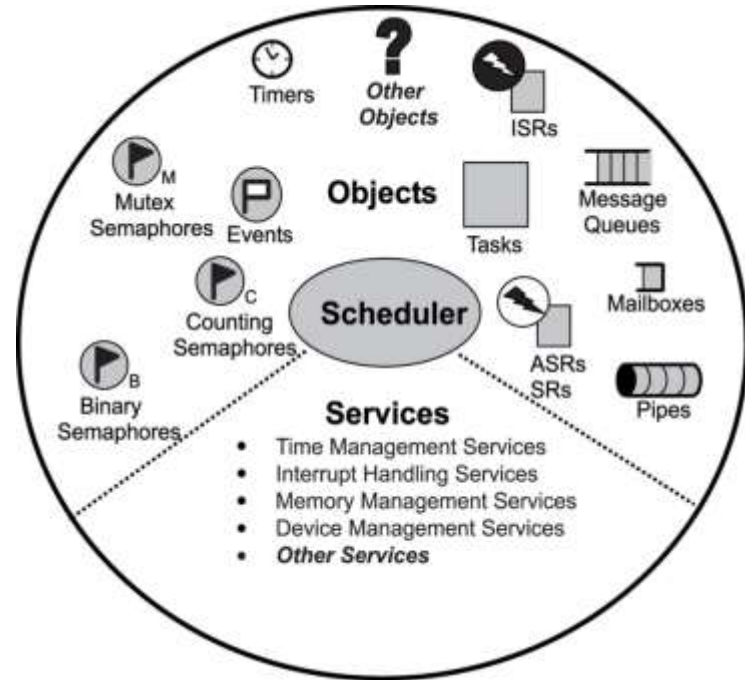
- References and Read more

RTOS Kernel

Kernel is the core component of any OS,

kernels components:

Scheduler is a set of algorithms that determines which task executes when.

Objects are special kernel constructs that help developers create applications for real-time embedded systems.

Services are operations that the kernel performs on an object
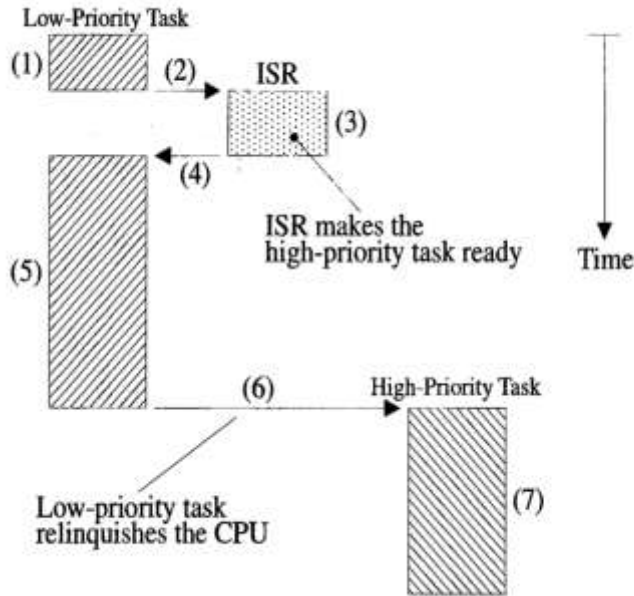
Priority Based Kernels.

- Are Kernels that decide which task will run regarding it's priority.

- Most real time kernels are priority based.

- Each task takes a priority based on it's importance.

- The Highest priority task is always ready to run.

- There are two types of priority based kernels:
  - Non-Preemptive Kernel.
 - Preemptive Kernel.

RTOS Task Status
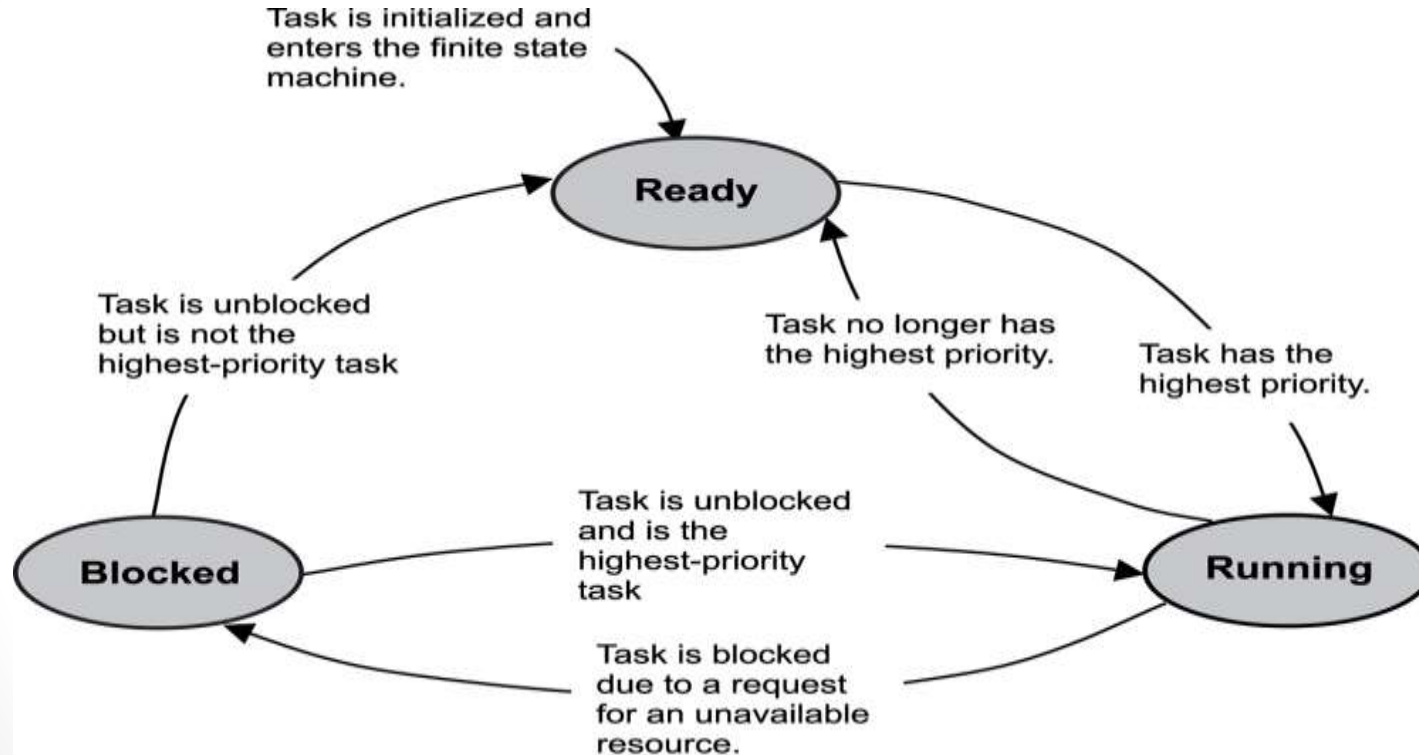
Non-Preemptive Kernel.

- An ISR Makes higher priority task ready to run.
- The ISR always return to the interrupted task.
- The new higher priority task will run when the current task gives-up the CPU.
- This minimize the data corruption risks in multitasking as each task finish before the other begins.
- Task response time will equal to the time of the longest task.
- Very few RTOS kernels now are using non preemptive kernels.

Low-Priority Task
(1)   (2)   ISR
(3)
(4)
ISR makes the
high-priority task ready   Time
(5)
(6)   High-Priority Task
Low-priority task
relinquishes the CPU   (7)

Preemptive Kernel.

- Here the highest priority task ready to run always given the control to the CPU.
- ISR makes the highest priority task run.
- The kernel next run the highest priority task in the ready queue.
- Here the response time to the highest priority task is it's best.
- Corruption of data may happen for non protected shared resources.
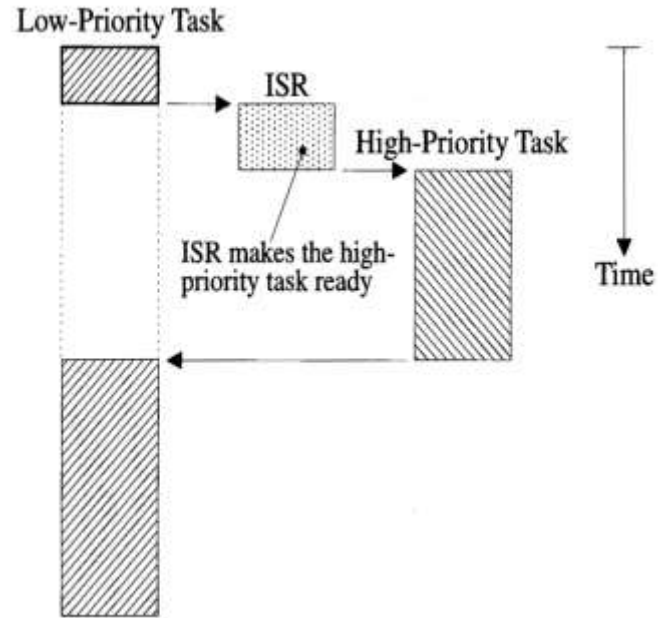
References and Read more:

- **Real-Time Concepts for Embedded Systems book** by Qing Li and Carolyn.
  - http://www.e-reading.club/book.php?book=102147

- **An Embedded Software Primer** by David E. Simon.
  - http://www.amazon.com/Embedded-Software-Primer-David-Simon/dp/020161569X

- **Linux Kernel Embedded Systems Building Blocks 2e** by Jean J. Labrosse.
  - http://www.amazon.com/Embedded-Systems-Building-Blocks-Ready/dp/0879306041

- **FreeRTOS website.**
  - http://www.freertos.org