

# Real Time Operating systems (RTOS) concepts

Abu Bakr Mohamed Ramadan

[Eng.abubakr88@gmail.com](mailto:Eng.abubakr88@gmail.com)

# Content:

- The Scheduler.
- What if two tasks have the same priority are ready?
- *Task object data.*
- System tasks.
- Hello World application using RTOS.
- References and Read more

# RTOS Scheduler.

- Scheduler the core Component of any RTOS kernel,
- Its a set of algorithms that determines which task executes when.
- It's keeping track on the status of each task, and decides which to run.
- In Most RTOSs the developers is the one who sets the priority of each task, regarding to this priority the scheduler will decide which task will run.
- The scheduler assumes that you knew what you where doing while setting tasks priority.
- A bad design for tasks priority, may leads to a high priority task hogs the processor for long time, this is called ***CPU starvation***.

# RTOS Scheduler.

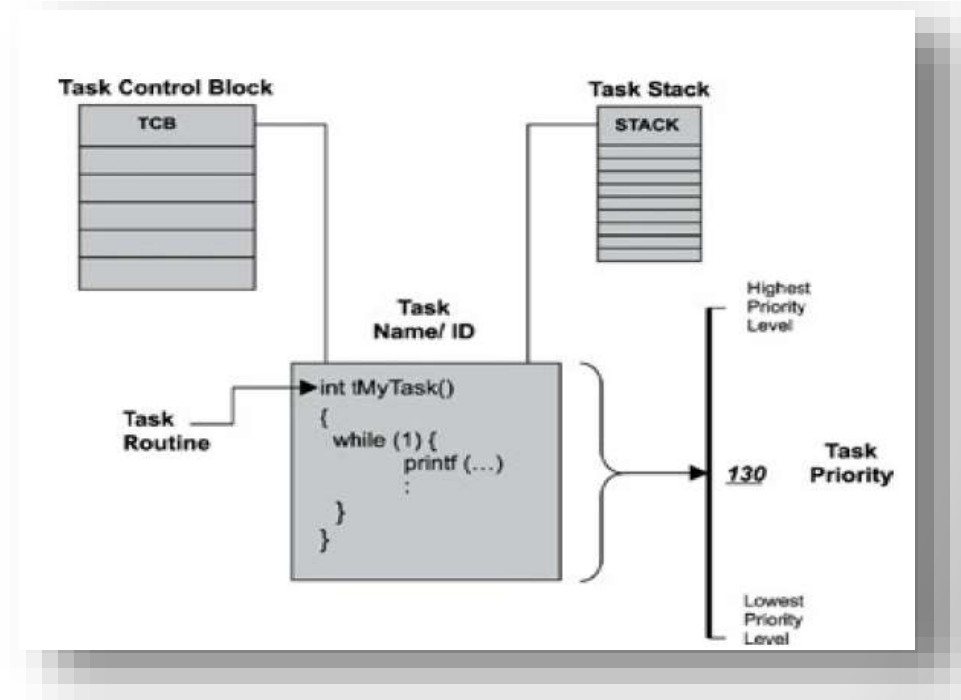
- It's keeping track on the status of each task, and decides which to run.
- Scheduler has no control on tasks on the blocked status.
- If tasks are blocked the scheduler waits an event to unblock this tasks, like an external interrupt from pushing a button.
- If no events happened, surely it's a bad design from your side.

# What if two tasks have the same priority are ready?

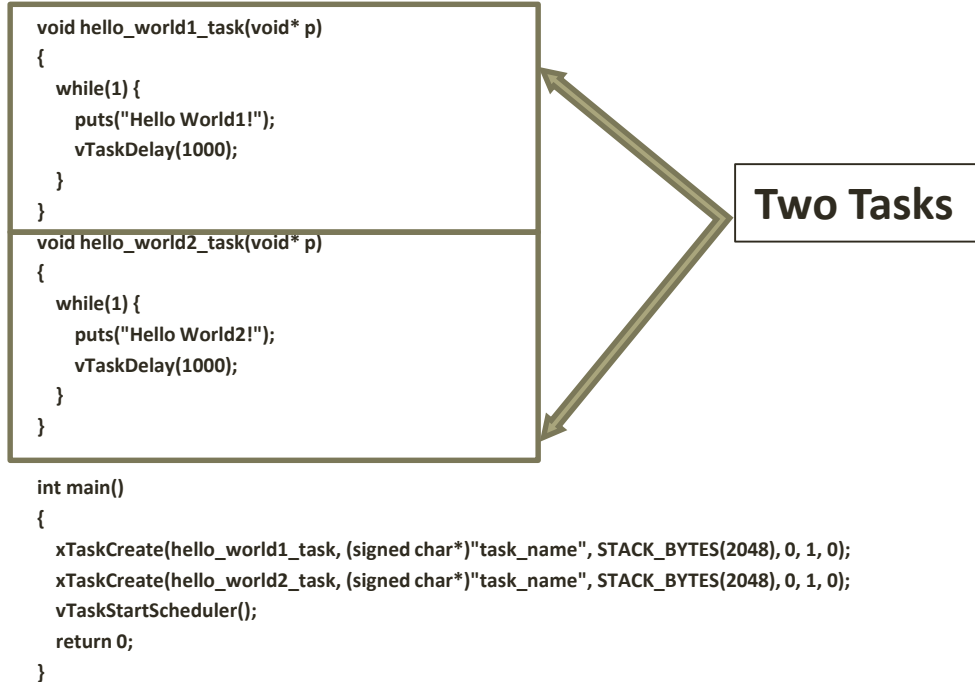
- Some RTOSs make it illegal to set two tasks with the same priority, and here the kernel limits the number of tasks in an application to the number of priority levels.
- Others will time slice between the two tasks(Round robin).
- Some will run one task until it blocks, then run the other task.

# Task object data

- Each task has an associated:
  - a name,
  - a unique ID,
  - a priority,
  - a task control block (TCB),
  - a stack,
  - and a task routine,



# Hello World application using RTOS.



# 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>