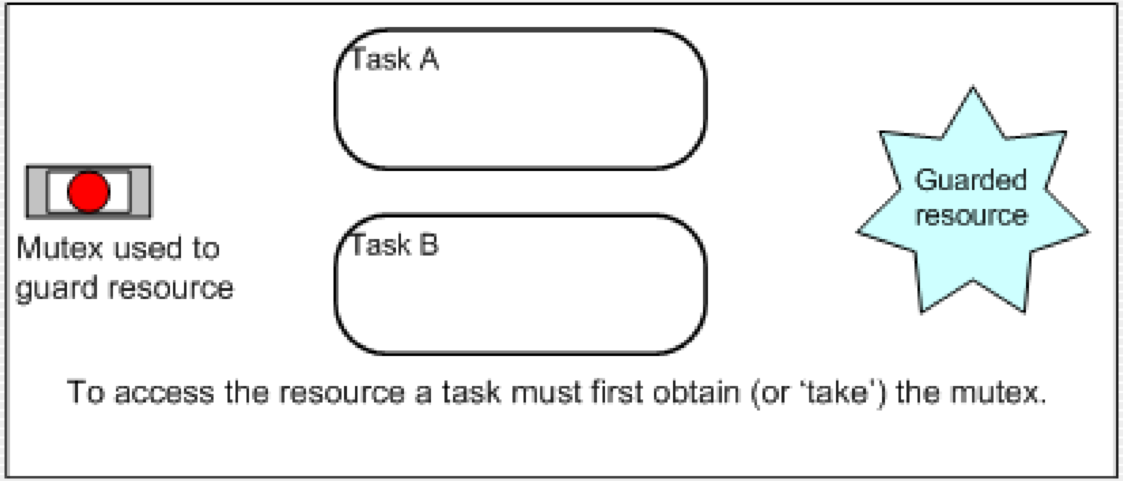
Real Time Operating systems (RTOS) concepts

Abu Bakr Mohamed Ramadan
eng.abubakr@gmail.com

Content:

Mutex

- *Mutex* is short for *MUT*ual *EX*clusion.
- Mutex is a special type of binary semaphore used for controlling access to the shared resource.


Task A
Task B
Mutex used to guard resource
Guarded resource
To access the resource a task must first obtain (or 'take') the mutex.

Mutex

- Mutexes include a priority inheritance mechanism to avoid extended priority inversion problem.
- This ensure the higher priority task is kept in the blocked state for the shortest time possible,
- Priority inheritance does not cure priority inversion! It just minimizes its effect in some situations.
- Hard real time applications should be designed such that priority inversion does not happen in the first place.

Mutex vs Binary Semaphore

- Ownership,
  - Mutex semaphore is "owned" by the task that takes it, so when a task locks (acquires) a mutex only it can unlock (release) it.
  - Binary semaphore has no owner, can be unlocked by any task,
- Usage,
  - **a mutex is a locking mechanism** used to synchronize access to a resource.
  - **Semaphore is signaling mechanism** (“I am done, you can carry on” kind of signal).

Determinism vs Responsiveness

- Determinism: determine the time that every task will be executed,
- Responsiveness: How your application be responsive to (External or Internal) events,
- RTOS Increase responsiveness and decrease determinism.

Why RTOS

- **Multitasking,**
  - Allows you to break a complex problem into simpler pieces so,
  - Focus on the development of each task rather than building a scheduler,
  - Increase the **Efficiency** of usage the CPU
- **Services,**
  - Kernel services are provided for resource management, memory management, event handling, messaging, interrupt handling , and more.

Why RTOS

- **Structure,**
  - Structure design of your application,
- **Portability,**
  - Your application will run on any platform the RTOS runs on.
  - So you can build an application for multiple targets from the same codebase.
- **Security,**
  - Some RTOS offers some security features, like encryption support for various communication protocols, memory protection unit (MPU) support,

Why RTOS

- **Debugging,**
  - Some IDEs (such as IAR Embedded Workbench) have plugins that show nice live data about your running process such as task CPU utilization and stack utilization
- **Support,**
  - Some RTOSs provides you with technical support and gives you access to the expertise of the engineers who developed each module, and they can advise and support you on your project.

Why not RTOS

- **Resources,**
  - RTOS will require extra resources,
  - Processing overhead due to context switch complex algorithms,
  - memory usage overhead due to OS source size,
- **Determinism,**
  - RTOS decrease Determinism,
- **Design**
  - Needs very carful design, and more developing skills,
  - It is never easy to port an RTOS
- **Debugging**
  - complex debugging (due to race conditions on resources shared among tasks)
- **Cost,**
  - Lots of RTOS needs expensive license, so product cost will increase,

RTOS Market

- 70% of Embedded Projects uses RTOS,
- 41% of RTOS projects uses a commercial OS,
- 34% of RTOS Projects uses open source OS without commercial support,
- 43% of commercial RTOS developers select it for technical support,
- These statistics from one of Atmel Corporation videos ,

Finally

- You can download course slides from the following link
  - https://www.slideshare.net/AbuBakrRamadan/presentations
- My Contacts,
  - eng.abubakr88@gmail.com
  - 01112979801
  - https://www.linkedin.com/in/abu-bakr-ramadan-18a6a850

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
- **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
- **Why RTOS**
  - **http://www.smxrtos.com/articles/whyrtos.htm**
- **Why Not RTOS**
  - **https://www.embedded.com/design/operating-systems/4403180/Get-by-Without-an-RTOS**