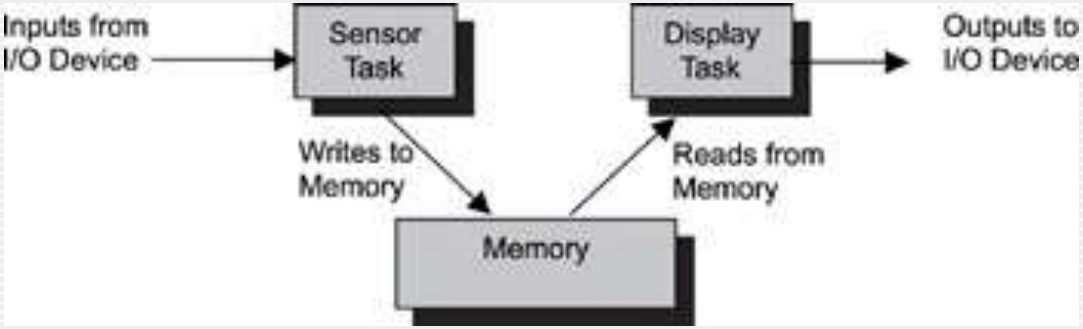
Real Time Operating systems (RTOS) concepts

Abu Bakr Mohamed Ramadan
eng.abubakr@gmail.com

Content:

- **Resource Synchronization**
- **Semaphore for Task Synchronization.**
- **Priority Inversion,**
- **Priority inheritance**
- **Race Conditions**
- **CPU Starvation**
- **References and Read more**

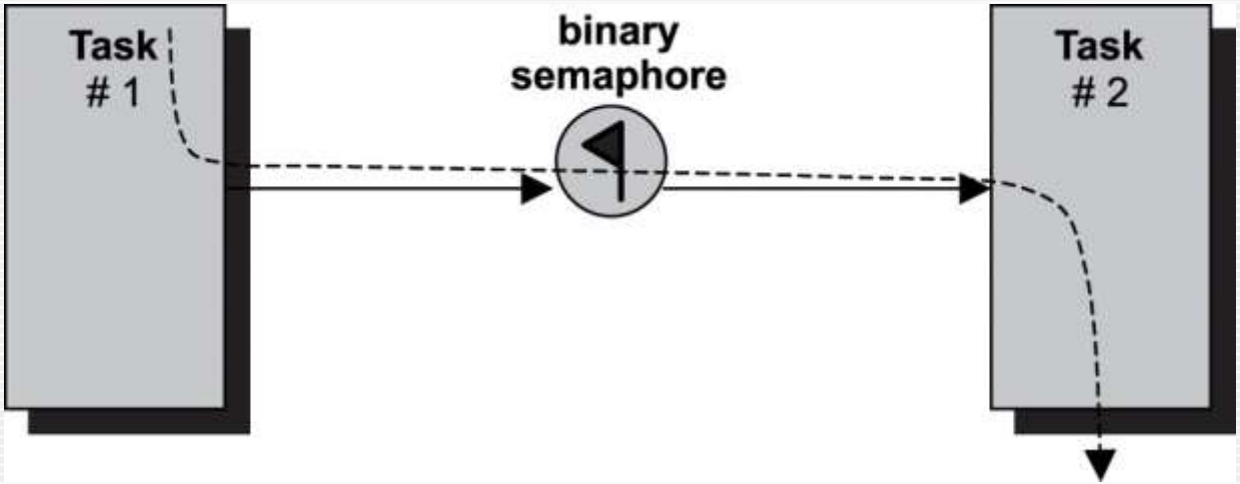
Resource Synchronization



- Determines whether access to a shared resource is safe, and, if not, when it will be safe.
- Access by multiple tasks must be synchronized to maintain the integrity of a shared resource.

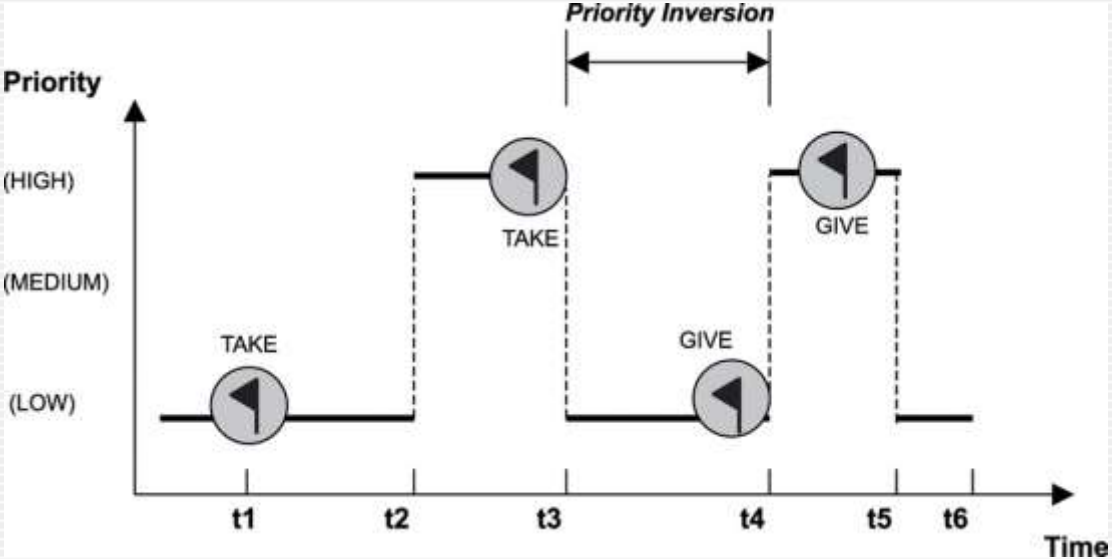
Semaphore for Task Synchronization.

- Semaphores are useful either for synchronizing execution of multiple tasks or for coordinating access to a shared resource.



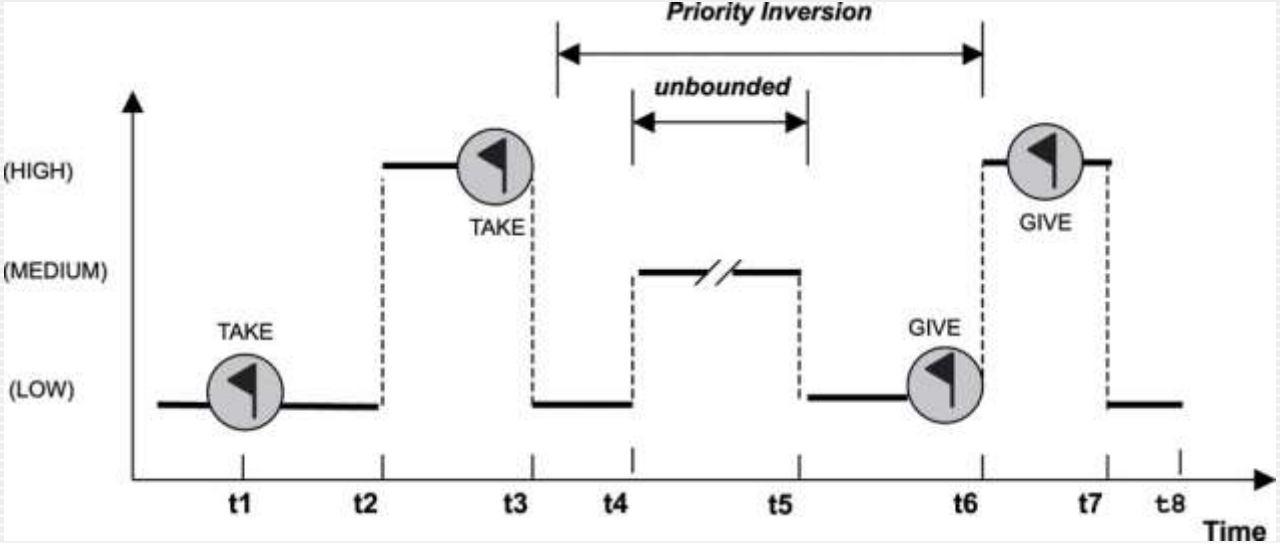
Priority Inversion,

- *Priority inversion* is a situation in which a low-priority task executes while a higher priority task waits.



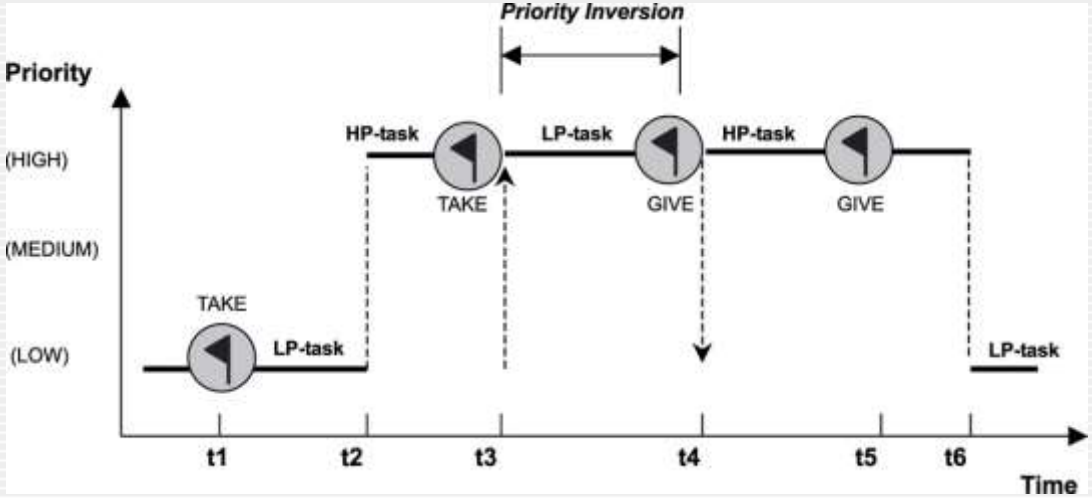
Priority Inversion,

- *Priority inversion* is a situation in which a low-priority task executes while a higher priority task waits.



Priority inheritance

- The Priority Inheritance Protocol is a resource access control protocol that raises the priority of a task,
- Static Priority and Dynamic Priority,



Race Condition

- when two or more tasks have access to a shared resource and they try to edit it at the same time,

```
if (x == 5) // The "Check"
{
   y = x * 2; // The "Act"

   // If another thread changed x in between "if (x == 5)" and "y = x * 2" above,
   // y will not be equal to 10.
}
```

Race Condition

- To prevent race conditions from occurring, you would typically put a lock around the shared data to ensure only one thread can access the data at a time.

```
// Obtain lock for x
if (x == 5)
{
   y = x * 2; // Now, nothing can change x until the lock is released.
              // Therefore y = 10
}
// release lock for x
```

CPU Starvation

- ***CPU starvation:*** occurs when higher priority tasks use all of the CPU execution time and lower priority tasks do not get to run.

- In a preemptive multitasking environment, If higher priority tasks are not designed to block, CPU starvation can result.

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