Lab 8 Real-time OS

Speaker: Hao-Yun Chin
Advisor: Prof. Tian-Sheuan Chang
May 4, 2003

uC/OS-II Kernel

- Basic jobs of uC/OS-II kernel
  - Task management
  - Inter-process communication
  - Interrupt handling

Context Switching

- Context Switching

  The process of swap in/out the context of tasks when scheduler choose a new task to run.

- Context

  Usually means values in registers used by the task.

- Switching Steps

  1. Save registers of current task to $stack_{(curr)}$
  2. Load new task's SP into CPU
  3. Restore registers of new task from $stack_{(new)}$
  4. Resume execution of new task

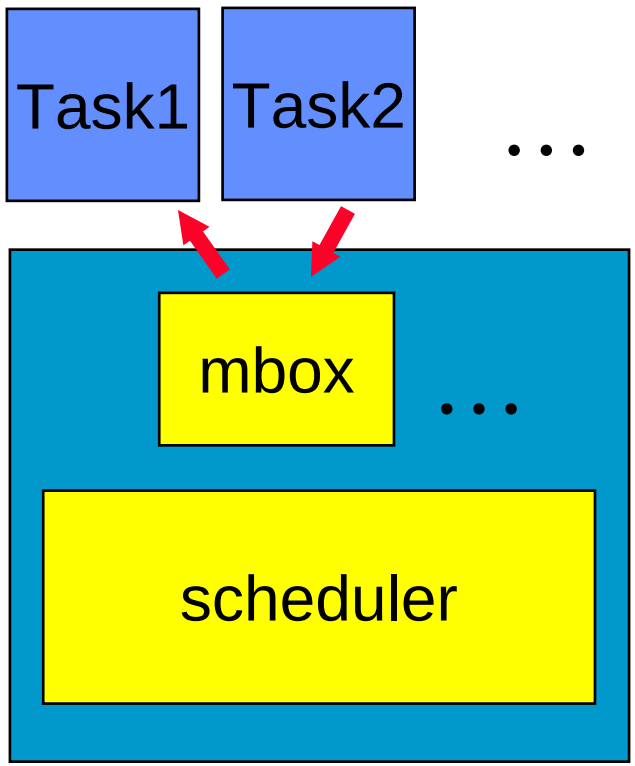
Events

- A way for Synchronization and Communication
  - Pend (Wait for an event)
  - Post (Signal an event)
- Events in uC/OS-II
  - Semaphore
  - Counting Semaphore
  - Message Mailbox
  - Message Queues

Events

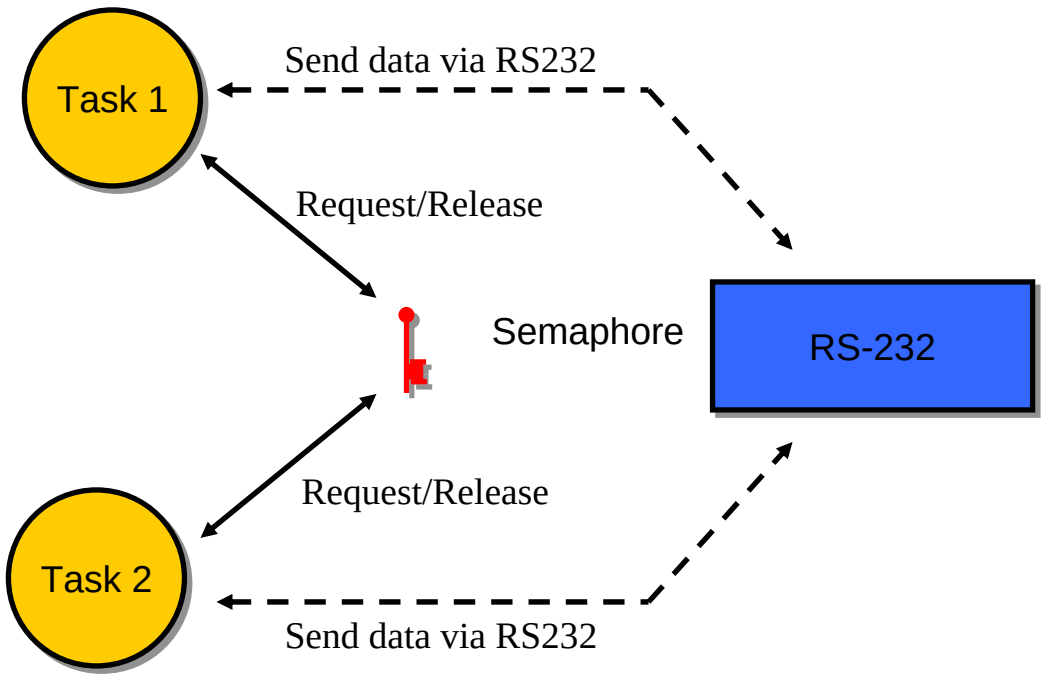
Priority: (H) 1 2 … (L)

```
Task1(){
    info = OSMboxPend(mbox);

}


Task2(){
    scanf(&info);
    OSMBoxPost(info);

}
```



Semaphore

- Semaphore serves as a key to the resource
- A flag represent the status of the resource
- Prevent re-entering Critical Region
- Can extend to counting Semaphore



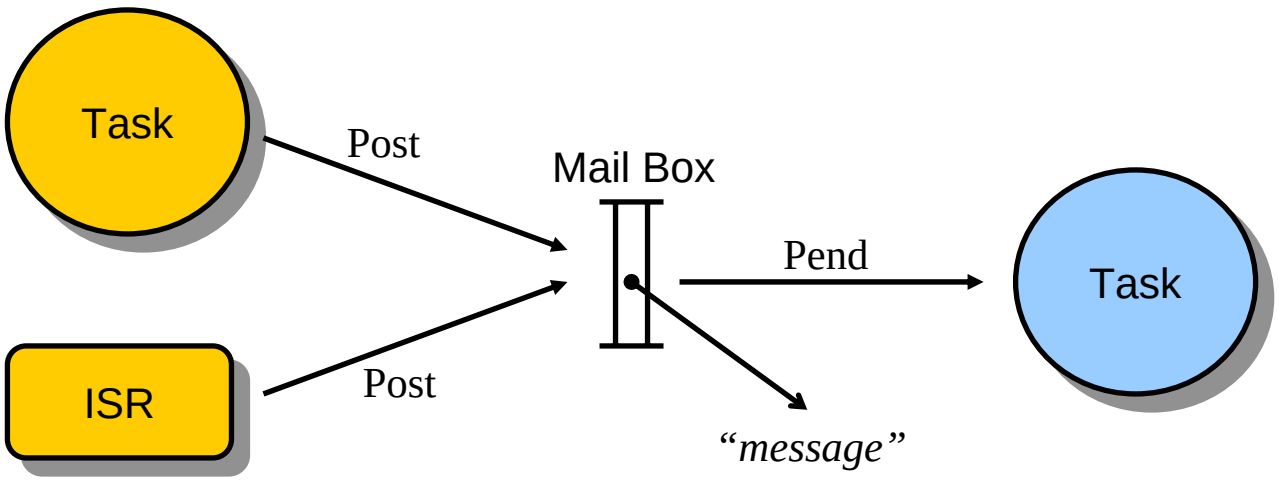
Semaphore

Using Semaphore in uC/OS-II

- OSSemCreate()
- OSSemPend()
- OSSemPost()

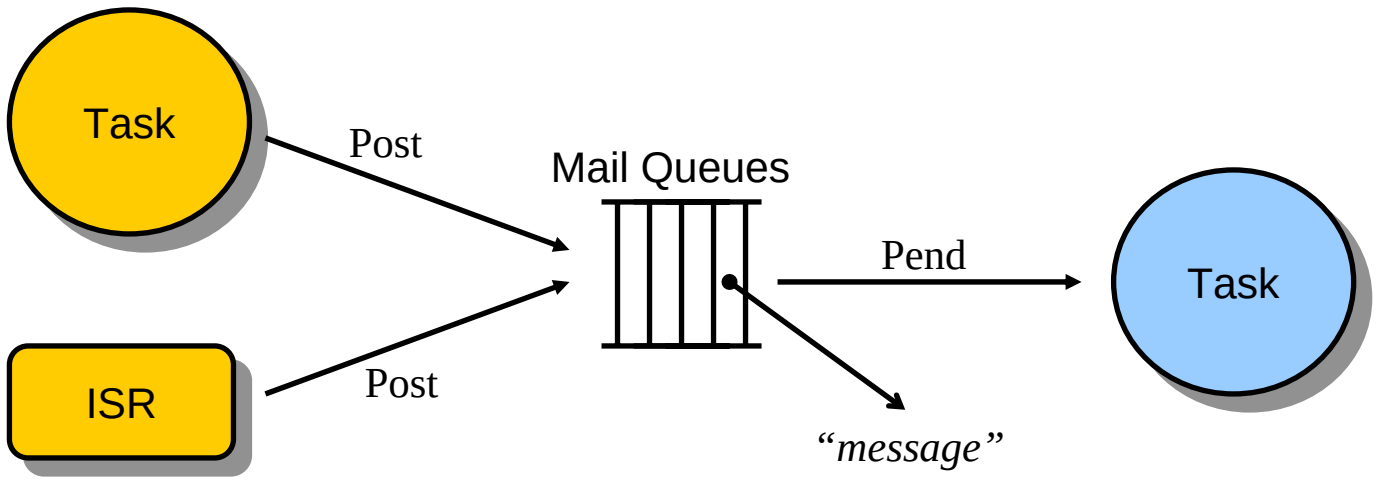
Message Mailbox

- Used for Inter-Process Communication (IPC)
- A pointer in MailBox points to the transmitted message



Message Queues

- Array of MailBox
- FIFO & FILO configuration



MailBox & Queues

Using MailBox in uC/OS-II

- OSMboxCreate()
- OSMboxPend()
- OSMboxPost()

Using Queue in uC/OS-II

- OSQCreate()
- OSQPend()
- OSQPost()
- OSQPostFront()
- OSQFlush()