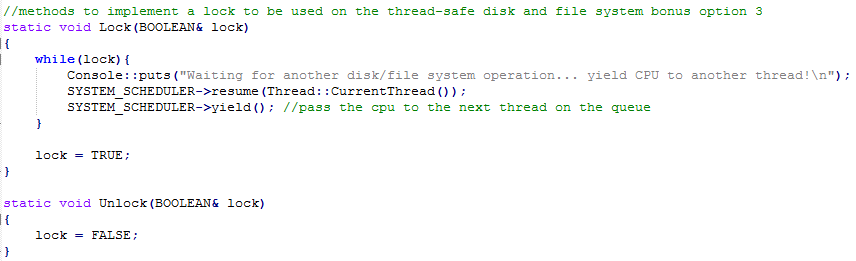
**CSCE 613: Project 6 Bonus Option 2 design**

**Student: Caio Duarte Diniz Monteiro**

## Date: 05/02/16

# Binary semaphore using Locks



The Scheduler class is responsible to manage the tasks allocation on the CPU. There are two scheduling policies implemented, first in first out (FIFO), and Round Robin. The desired policy is chosen by manipulating the defined value **SCHE\_ROUND\_ROBIN**, when it is defined, round robin is used, otherwise it uses FIFO.

# Disk access

Interrupt class had to change its dispatch\_interrupt function to deal with the round robin scheduling. When using round robin scheduling the timer will trigger a context switch through an interrupt, so we need to let the PIC know that the interrupt has been taken care of before switching contexts on the handle\_interrupt function.



# File access

Thread class maintains the TCP of the existing threads on the system. To allow thread termination, an extern pointer to the system scheduler was included, so, upon thread\_shutdown, the system scheduling terminate function is called. To properly work with the round robin scheduling, interruptions were disabled when setting the context of the thread, enabled upon thread start, disabled right before the context switch and enabled right after.



# File System access

Behavior of the SimpleTimer is dependent on the defined variable SCHE\_ROUND\_ROBIN. If the OS is using round robin, then it needs to perform a context switch every 50ms, otherwise it keeps the previous behavior of simply displaying a message every one second. The changes are made on the handle\_interrupt function and include divide the frequency of the timer by 20, to obtain the 50 ms, put the current thread at the end of the scheduling queue, and yield execution to the next available thread.