

CS404 - Real-Time Operating Systems

Lab Assignment #5

Spring 2009

The purpose of this assignment is to add counting semaphores to the UIKAPI. This assignment can be done in groups of up to two people.

You are to implement the following counting semaphore functions:

- `UIKSem* UIKSemCreate()`; - will create and initialize a new counting semaphore. The new semaphore should be initialized to 1. The function should return a pointer to the new semaphore, or a value of -1 if there was problem creating the semaphore.
- `void UIKSemPend(UIKSem* sem)`; - will perform the $P(s)$ (wait or test or acquire) operation on the semaphore. This function should decrement the semaphore value, and if the result is negative, should block the requesting task.
- `void UIKSemPost(UIKSem* sem)`; - will perform the $V(s)$ (signal or release) operation on the semaphore. This function is the one that should be called when a task is leaving a critical section. It should increment the semaphore value, and if the result is less than or equal to zero, should unblock the highest priority task that is waiting on the semaphore by changing its state to ready.
- `INT8 UIKSemValue(UIKSem* sem)`; - will return the current value of the semaphore.

In addition to implementing the semaphore functions, design a set of tasks that demonstrates the proper operation of your semaphore functions.