

Problem Set 7

Semester 1, 2012/13

Due: November 11, 23:59

Marks: 5

Problem 1 [5 marks, submit to PS7P01]

Consider the following naive implementation of concurrent execution of two processes in Python:

```
def task1(x):
    reg = x[0]    # increment a shared variable
    yield         # simulate the non-atomicity of load and store operations
    reg = reg+1   # that would be observed in realistic hardware
    yield
    x[0] = reg
    yield

def task2(x):
    reg = x[0]    # increment a shared variable
    yield         # simulate the non-atomicity of load and store operations
    reg = reg+1   # that would be observed in realistic hardware
    yield
    x[0] = reg
    yield

def scheduler():
    x=[0]
    tasks = [task1(x),task2(x)]
    for i in range(6):
        tasks[i/3].next()
    print "x=",x[0] # will print "1" due to interfering increment
```

The generators `task1` and `task2` simulate two very simple tasks which increment a shared list element non-atomically (*why does it have to be a list element and can't just be a simple variable?*). The procedure `scheduler()` simulates a (rather limited) round-robin scheduler. The simulation of the concurrent execution can be started by just calling the `scheduler` procedure. The result will be the “interfering” incrementing of list element `x[0]`, which will have a final value of 1.

Based on this idea, create a more sophisticated concurrency simulation system that can also block tasks. Then, add semaphores to your system, and showcase their use by wrapping the non-atomic increment operation on `x[0]` between the `acquire()` and `release()` of a semaphore.

Be sure to explain, in comments added to your code, how your system works, and how it is expected to be used. Add a main program that demonstrates the use of your system.

Hints: you will need to add task identifiers to your tasks (it is ok to have these identifiers as arguments to your semaphore procedures). Moreover, the semaphore data structure will have to be, for reasons of sharing, a list element, and not a simple integer.

Your submission should consist of a Python file containing your simulation.