

University of Puerto Rico Rio Piedras Campus

Department of Computer Science

Kai-Ming Chow Benabe

CCOM-4017-0U1

Prof. Jose R. Ortiz

Assignment 2: Threads and Scheduling

This assignment consists of two programs that simulate a consumer-producer problem with the use of a server and multiple client connections. The clients send multiple messages to the server that contain the amount of time to run a job in the CPU. These messages are sent to the server with the use of UDP sockets. The server works as a scheduler and runs two threads: a producer that listens for incoming messages and stores the jobs in a shared Queue, and a consumer that retrieves the jobs from the Queue and executes them. The server also relies on the use of semaphores to synchronize the jobs that run in the CPU and mutex locks to protect shared data.

Significant Variables:

Sock: UDP socket.

N: amount of jobs that the server runs.

Mob_id: mobile client identifier specified by the user.

Cpu_time: randomly generated time a job will spend sleeping.

Data_split: array that contains mob_id and cpu_time.

Buff: an implementation of an unsynchronized Queue.

Total_time: dictionary with the id of each mobile client as the key and the total time spent on the CPU as the value.

Instructions:

1. Run the server program by executing the following command on the terminal:

```
python scheduler.py <server port>
```

2. Open multiple tabs on the terminal to run various mobile client programs by using the command:

```
python mobile.py <mobile id> <server address> <server port>
```

3. Wait for the jobs to complete for the server to output the the amount of CPU time that each mobile client consumed.

Issues:

Had discussion with Alejandro Deloach about the implementation of an unsynchronized Queue.