CS2S563 Operating Systems

Semester Two, Week 7 Tutorial

Access the code for this tutorial on github if needed:
https://github.com/alunkingusw/OSCTutorial6

Tasks

1. copy the week18UDPGameServerThreaded.py from the last lecture and run it

2. copy the week18UDPGameClient.py from the last lecture and run two instances, either twice on the same machine, or on two different computers (make sure you configure the IP address to connect to the server correctly)
    a. Does it crash?
    b. See the demonstration on the board if not, and modify the code to simulate the problem

3. Declare a new semaphore at the top of the program with a count of 1. For example, the line below creates a semaphore called mutex.

```
mutex = threading.Semaphore(value=1)
```

4. Identify the critical areas where you need to prevent threads accessing shared memory and surround them with `mutex.acquire()` and `mutex.release()`. Can you get the server to execute correctly?

5. This problem is mainly because of modifying the data structure. Reading the data doesn't cause a problem. So it's a simple version of the readers and writers problem. Have a think about how you might solve the more complex version of the readers and writers problem using semaphores
    a. the readers and writers problem is defined as follows
        - only one writer can enter a section of code at a time
        - if no writer is in the section of code then any number of readers can enter at a time
        - a writer cannot enter until all readers have exited the section of code