CS 330: Operating systems

# lab 12: FILE Management Server

# group members:

### Abdullah Waris (270745)

### Faraz Ahmed (260267)

### shaheer ahmed khan (265812)

# UI HANDLING

## Running the code:

You can run the server file (server.py) and then run the client file (client.py).

You can then go on following the instructions of the UI.

# Task 1: Implementing a queue for readers

### Your task is to make sure that while the file is being read no writer is allowed while maintaining the order of writes as discussed in class.

Order of writes maintained in a queue. This was done by saving the threadID’s in a queue and popping from the start and adding to the end.

### Any request to open the file in write mode is considered to be in active writing till the user closes the file.

The way I’ve accomplished this is by simply adding a semaphore that is set a write semaphore to 1 and acquire it.

### SIMILARLY, any request to open the file in read mode is considered to be in active read till the user closes the file

Just like the above solution. I have used a semaphore and set it to the required constant. And whenever a file is reading it. It acquires the semaphore.

### Multiple users can read the file concurrently but writes will be mutually exclusive.

If follows from the above two solutions. As the semaphore for write is set to 1 but for reading it is set to a predefined constant that can be more than 1 and hence allowing for a file to be read concurrently.

# Task 2: implement a limit on the user name to access a file.

### Multiple users can access your system.

Multiple users can access my system. Infact the server is in a infinite wait loop catching incoming calls to connect.

### Though it is not required that you implement any type of security, but any user may not access more than 5 files.

A user cannot have more than a constant specified amount of logins. Also the user can not access more than a constant number of files.

### If more than 5 requests are placed then the requesting thread must wait.

This is accomplished using semaphores on the user.

### Each file can only be access by 3 users, be it for read or write.

This is accomplished by using semaphores on files.