**Programming Assignment #3**

**CSCI 4534**

**Date Assigned**: March 27th

**Date Due**: Thursday, April 5th 11:59pm to Blackboard

**Problem**:

Create a CPU scheduler that runs a round-robin algorithm. Each client sends the server an integer named **burst** (that represents a CPU burst) and the name of its private FIFO through the common FIFO, named **commonFIFO**. The common FIFO is made only one time, and will be opened for reading only one time before processing any client data. The server is asked for a number of clients and an integer time quantum named **timeQuant**. All processes arrive at the Ready Queue at time zero and are enqueued in the order they are submitted.

The server receives input from each of several clients. The server reads the input integer from a client, creates a node (a PCB) for the client, then enqueues the PCB to a queue named **Ready**. This is done for each client. Display the contents of the queue periodically.

After all client information has been read and stored in the nodes of the queue, processing can begin. The process represented by the process at the head of **Ready** will “run” until the process completes or the time quantum expires. If the process completes, the client will be notified of the completion time and the process is removed from the system. If the time quantum expires, the node will be enqueued to **Ready** and the remaining time for **burst** is stored back into that node (its PCB). The clock can be manipulated by adding one to the clock to represent handling one CPU burst at a time.

A **clock** value (a counter) records the current time.

When a client’s burst is completed (equals 0), the value that is in **clock** is recorded, a **turnaround time** is calculated and both the clock value and the turnaround time are sent back to the client through its private FIFO. The server also reports when a client completes and the current clock time. When **all clients** have completed processing, the server reports the **average turnaround time** for the set of jobs.

**Server Note:**

* 1. **commonFIFO** is created and opened once.
  2. When all clients have completed and results have been displayed, the server shuts down.

**Client Note**: The client program is one program that is run multiple times, thus, you will **open multiple windows** that run the client program.

**For 5 points extra:** Each client includes an arrival time along with its burst and private FIFO name in the struct sent to the server. Arrival times may be given in increasing order.

**Submit your work to Blackboard as follows:**

* Create a folder named lastNameFirstInitial\_CSCI4354\_Program# e.g. HallS\_CSCI4354\_Program3
* Place the following into the folder

1. Your **.c** files. The .c files of your client and your server should be **documented** following the Programming Guidelines. Clearly indicate which program belongs to the client and which belongs to the server.
2. In your documentation, add an **estimated** and **actual** time spent for each of the following: design, implementation, and testing.
3. A separate **ReadMe** file that explains how to execute your programs.
4. Screenshots of execution showing **Input values** and **Output values**.
5. **Gantt chart** for a tested set of clients

* Zip the folder: Right click on the folder and send to compressed file

**Late Programs**: Programs are due to Blackboard at 11:59 pm on the due date. Programs turned in after the due date will be penalized 20 points per 24 hour period up to 2 days late.

**Programs that don’t execute properly on sapphire**: Programs that don’t compile receive a grade of zero.

**Back up your files often!**

Homework for Tuesday, April 3rd:

Turn in a **preface** for your client and your server, which contains all parts designated by the Programming Guidelines.