**Programming Assignment #2**

**CSCI 4534**

**Date Assigned**: Tuesday, February 20, 2018

**Date Due**: Tuesday, March 6, 11:59pm to Blackboard

**Problem:**

Multiple clients send the name of one job and its memory request (number of bytes) to the server. The server acts as a memory manager and allocates memory for as many clients as possible using paging as its memory allocation scheme.

Assume that pages and frames are both of size 128 . Determine how many frames are required for each memory request. A main memory will consist of at least 16 frames. If there are sufficient frames available, “assign” pages to frames in order, starting with frame zero. That is, if client 1 has a memory request for 320 bytes, it will need 3 frames. Page 0 is placed in frame 0 from physical address 0 to 127, page 1 is placed in frame 1 from physical address 128 to 255, and page 2 is placed in frame 2 from address 256 to 319. Next, client 2 pages are placed into frames beginning at frame 3, starting address 384.

If the available memory is not sufficient for the next requesting job, a message saying that there is not sufficient memory is sent back to the client. If the request is invalid (such as zero, a negative number, or a request larger than the total memory size) the program will respond with an appropriate error message. Make sure that you test for each of these eventualities. Assign memory addresses to each page according to the frame to which it is assigned. Calculate the starting and ending physical addresses of each page and the amount of fragmentation in the final page.

The server begins by asking for the number of clients and the initial size of memory in terms of frames.

A client contacts the server only one time! The client sends the server a memory request, the job name, and the private FIFO name through a **common FIFO** (what we called FIFO\_to\_server) that goes directly to the server. The server responds to each client through their **private FIFOs** with the frame numbers that this client was assigned, the starting and ending physical addresses and the amount of fragmentation in the client’s last page, or an appropriate error message. Use at least 3 clients.

After all clients that fit have been assigned memory, show a ‘**map**’ of the allocated memory on the server side.

**At least three clients are expected in the system. Each client:**

* + - 1. Prompts for job names and a memory request from the user.
      2. Sends the name and memory size for each client, and a string that represents the name of a private FIFO (from which it will read the result from the server) to the server through a common FIFO that is used by all clients.
  1. After receiving a reply from the server, the client displays the results sent from the server.
  2. The client will terminate.
  3. **Note**: The client program will be one program that is run 3 times, thus, you will open multiple windows that run the client program. The minimum number of clients must be 3 and the maximum can be your choice. You will use one common FIFO that is created and opened by the server only one time.

**Server:**

* 1. Prompts for and receives an integer that determines the number of clients (from the keyboard at the server, and this will be at least 3) and total memory in terms of number of 128 byte frames.
  2. Receives job names, memory requests, and private FIFO name from each client through the common FIFO.
  3. Acts as a memory manager, assigning pages to frames for clients.
  4. Calculates internal fragmentation in each client’s final frame.
  5. When all clients have completed, the server shuts down.

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

* Save your client and server programs using Notepad++.
* Create a folder named lastNameFirstInitial\_CSCI4354\_Program# e.g. HallS\_CSCI4354\_Program2
* 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. Please put this in Notepad++ also.
4. **Input values** and **Output values** with the screenshots of execution.

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

**FIFOs**: Use at least 4 FIFOs for this assignment, the commonFIFO that all 3 clients write to, named common\_FIFO, and a private FIFO for each client.

**Late Programs**: Programs are due to Blackboard at 11:59 pm on the due date. Programs submitted late will have points deducted.

**Back up your files often!**

Homework for Tuesday, February 27th

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