Programming Assignment 2 – Preface

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
\* **Programmer**:  Ethan Hann \*  
\* \*  
\* **Course**:  CSCI 4534 \*   
\* \*  
\* **Date**:  October 15, 2017 \*

\* \*  
\* **Assignment**:  Programming Assignment #2: Client/Server Memory Manager \*  
\* \*  
\* **Environment**:  Notepad++ and Sapphire UNIX server \*  
\* \*   
\* **Files Included**: memManager.c \*  
\* \*  
\* **Purpose**: To simulate a memory manager on a server that uses paging as its memory \*

\* allocation scheme \*  
\* \*  
\* **Input**: From ‘x’ number of clients: the name of one job and its memory request size \*

\* From server: the number of clients and the initial size of memory in terms of frames \*  
\* \*  
\* **Preconditions**: All pages and frames are of size 256; main memory consists of at least 12 \* frames  
\*   
\* **Output**: To each client: the frame number(s) that the client was assigned and the amount of fragmentation generated from assignment; or an error message if couldn’t assign any memory

Server output: a ‘map’ showing the allocated memory  
\*   
\* **Postconditions**: None  
\*   
 **Algorithm:**

*Server:*

Define a struct for output:

* fragmentation (integer)
* frame numbers (int array)

Create commonFIFO

Open commonFIFO for reading

Ask user for number of clients and initial size of memory (in frames)

For each client:

Read from the commonFIFO

Create clientFIFO

Open clientFIFO for writing

Calculate number of frames needed

Calculate fragmentation

Mark allocated frames busy

End For.

*Client:*

Define struct for input to server:

* job name (string)
* memory request size (integer)
* private FIFO name (string)

Define struct for input from server:

* fragmentation (integer)
* frame number(s) (integer array)

Get job name and memory request from user

Assign struct values to input values from the user

Open commonFIFO for writing

Write the struct defined above to the commonFIFO (*input to server*)

Open privateFIFO for reading

Read output from server from privateFIFO into another struct (*input from server*)

Display the results from the server to the user nicely

Close privateFIFO

Terminate client