Anthony Berton
Tyler Groom
Titus Sudarno
Thomas-James Le
CPSC 351 Section #2

Design for Assignment #2

Project Purpose

This project is a simulator that will read the contents of an input file to simulate the execution of processes and the actions made by memory management. The program will trace these events and the memory map and the input queue status of these events using an output file. At the end of the simulation, the program will output the average turnaround time. The events that are traced are the process arrival, the process' admission to the memory, and the completion of the process. The simulation is programmed entirely in C and will run on a Linux OS. The program consists of the main.c file as well as the header file prototype.h which contains the functions used in main. The header file also references three other files; process.h, queue.h, and memory.h.

Main.c

The main file will first get the user input using the getInput function. These inputs include the memory size and the associated parameters. The program then uses those inputs and puts them in a process list. A queue is then created with a capacity equal to the number of shared processes. Finally, a shared framelist is created using the function createFrameList. The main function then calls the mainLoop function. In the mainLoop function, a long type variable called time is declared and initialized with 0. A while loop is then created which performs three actions; it adds the new processes to a queue, terminates the completed processes, and then assigns any available memory to a process. Afterwards, time is incremented by one. The loop will repeat until either the current time exceeds the maximum time or the queue is zero and the frame list is empty.

H files

The memory file performs several different actions. It creates a frame list which contains the page size and the number of frames in the list. It also checks if the processes fit into memory. If so then it fits them into the memory. It also checks if the framelist is empty and if there is available memory.

The queue file adds and removes processes from the queue. The file can also search for an index in the queue and remove the process in that index. Lastly, it is able to iterate the queue index. The queues are created using a struct object which consists of variables for the elements, the capacity, the size of the queue and the front and rear indexes.

The process file only contains one struct object which stores information related to process. These include the process id, the arrival time of the process, the lifetime, the required memory, the amount of time added, whether or not the process is active, and the time the process finishes.

Errors/Shortcomings

We ran into one error when we ran the program. It is related to the end of file (eof) command in the main.c file.