Assignment 3 Report

Author: Patrick Fleming
Student Number: C3253586

Testing

The testing conducted was to first test large and small frame numbers. This caused the results to error if the number of frames was smaller than the number of processes because for locally assigned the frames are never reassigned so the processes without a frame cannot run because of this added check to assure that frames are greater than processes. After this different time quantum's were tested to ensure that the scheduler was correctly using round robin. This was tested using large and small numbers and comparing the output times and fault times.

Next, I tested the schedular to make sure that the amount of pages cannot exceed 50 for one process and that it does not matter the amount of instruction pages the schedular runs correctly.

Results

The results show that if you have more frames then what is needed for each process both global and local replacement run the exact same. If you have a large time quantum but low frames the benefit of a large time quantum isn't seen as each process will fault before the time quantum is finished. Having processes that complete the same instruction over and over again make it so that local replacement work better than global because you end up replacing the instructions other processes need causing more faults to occur. Overall the results show that global is better for a smaller amount of frames especially if the frames are less than the amount of processes and locally is better for processes that repeat the same instructions over and over again.

Edge Cases

One edge case considered is if the amount of frames is less than the amount of processes if this happens local replacement doesn't work so the program will throw an error and not run.

Another edge case was if the time Quantum was larger than the amount of instructions a process has, this would cause processes that repeated the same instructions to finish quickly but other processes to finish slightly later than if the time quantum was smaller than the amount of instructions.

The edge case of a process having more than 50 instructions was considered and causes the program to throw an error as this is not allowed.

Review

Because I decided to implement this in the most stupid way possible it caused me a lot of issues in getting the timing of faults and the turnaround time correct.