Report Assignment 3 COMP2240

Scott Lonsdale (C3303788)

How you tested your program

Most of the main testing was done by using an output to the cmd line of the workflow of each process, showing when each process was blocked and unblocked. Also having a string output for the queue on each loop of the main function. This allowed me to see any deviation to the correct work method as presented in the supplied pdfs. Doing this help fixed a number of issues surrounding my code such as starting a page fault taking one time unit. I also used cmd line outputs inside the replacement functions to see if the correct processes were being replaced in memory which allowed me to fix some dumb runtime errors such as calling i instead of lastUpdated[i] in comparisons.

Comparison of the replacement methods

Comparing the two replacement methods based on the provided inputs we can see that LRU is shown to have the faster turnaround time of the two policies. However, this may be biased as we are only using one dataset, and that the two policies are very close together in optimised performance. We can see this is the case, as the clock policy has one more fault compared to the LRU policy.

Interesting observations

Compared to the first assignment, I ended up using an actual priorityQueue instead of just determining the next process using a function which made the structure of the code alot easier to develop. However, I probably should have used a circularly-linked linked list for the clock policy as the current solution is a bit hacked together. Another big error that I had for a long time, was that the process output didn't reset when I started a new simulation as they were stored in the actual process classes. I probably should have implemented a separate output class to remedy this, instead of using the current workaround.