**Virtual Memory Manager Algorithm Analysis**

* *How do you handle a page fault?*

The virtual memory manager algorithm that I’ve implemented follows the concept of “Least Recently Used” (LRU) victim selection so that the frame storing the page that was used least recently will be the victim that is selected to be replaced when SAC-SimOS decides that a new page needs to be written to memory. Specifically, when a page fault occurs, the frame containing the page used least recently will be written to the hard disk (SimHDD) and then that frame will be given permission to subsequently be overwritten by memory in order to store the newly requested page.

* *How are you aligning your paging strategy with locality?*

By utilizing a “Least Recently Used” victim selection algorithm, I’m able to exploit SAC-SimOS’s temporal locality because only the pages least recently used by the system will be replaced. This allows the system to more quickly access the resources that it needs, which are also most likely the resources that it has also used most recently, and generally decreases the frequency of page faults occurring.

* *When do you allocate another block on the disk?*

I allocate another block on the disk whenever HDD is full and there is still a frame in memory that needs to be overwritten because SAC-SimOS wants to add another page to memory. By writing a new block on the disk, I am able to preserve the state of the old page that was being used by memory so that it can then be used again at a later point in time if SAC-SimOS were to ever need it. It is only when HDD is full and there are no available blocks to store a process in that I add another block to the disk.