Intoduction:

The objective of memsim is to simulate the actions a single level virtual page table using different page replacement policies. The policies used in memsim are Optimal, Least Recently Used, and Clock. Optimal looks ahead in what is called a reference string, a list of every memory access that occurs, and selects pages that are used the farthest away to remove. Least Recently Used uses a policy fitting of its name; it removes the page that was accessed furthest from the current time. The Clock policy iterates through the pages that are currently loaded in frames. If the current page is not already loaded, the policy will begin marking a use bit for replacement, If the clock reaches a frame where the page is marked as not being used it will be evicted.

Optimal Replacement Policy

The first approach the group made to implement the Optimal Replacement Policy turned out to be to complex. The initial approached used a structure that contained a Boolean number to show if the trace was designated read or write, an unsigned integer variable to represent the virtual page number being loaded into a frame, and an array that stored the index of every occurrence of that unique virtual page number. The program would then create a linked list between each unique virtual page number structure, and would iterate through them following the policy outlined in the textbook. The complexity of this type of structure, and populating its values proved to be a waste of resources and troublesome to comprehend. The new approach was to create a structure that contained each virtual page number as an unsigned integer, and a character variable to represent if it had read or write access. Two arrays of these structures were needed to implement this policy. One array (pm[]) represents the physical memory, and its size is determined by the number of frames, decided by the user. Another array (refrence[]) is represents the reference string of every access to memory in the trace file. This list is iterated through