

- 1) List what you have done and how you tested them.
  - a) I have completed implementing the functions MMU, get\_PFN, pagefault\_handler, fifo, lru, and clock. I have tested all of them by running my program with different input files and comparing the output with the vm\_reference program by doing a diff on the output.
- 2) Explain your design of data structures.
  - a) Most of the design of the data structures applies to the page replacement algorithms functions.
    - i) FIFO: Pages to replace are kept track by using a pseudo queue. Since the indexes are the same as the PFN, the function uses an integer, called fifo\_tail, to determine which PFN to return. First, the function will copy the value of fifo\_tail to a variable. Then it will increment by 1 then mod by MAX\_PFN. Lastly, it will return the previous variable.
    - ii) LRU: The data structure used to implement this function is a double linked list. When the program starts, all of the miss will call the pagefault\_handler function. That function will initialize the linked list by adding all of the free frames. Afterwards, when there is a hit, that page will be moved to the top of the linked list. If there is a miss, the bottom page will be removed and its PFN will be returned. Also, the new page will be placed at the top of the linked list.
    - iii) CLOCK: There is an array that keeps track of all of the reference bits. First the program will initialize all of the bits to 0. If there is a page hit, the function will change the page reference bit to a 1. When there is a miss, the clock function will loop through the array. If the bit is 1, change the bit to 0 and move to the next index. If the bit is 0, return the PFN.