**Here is pseudo code of what your Virtual Memory Manager might look like. Refer to the enclosed C files for file operations in C, address translation, and extracting a virtual page number from a virtual address.**

//Virtual Memory Manager

#define TLB\_SIZE 16

#define PAGES 256

#define PAGE\_MASK 255

#define PAGE\_SIZE 256

#define OFFSET\_BITS 8

#define OFFSET\_MASK 255

#define MEMORY\_SIZE PAGES \* PAGE\_SIZE

//The TLB structure

Declare a 'tlb' array of size TLB\_SIZE

Each element of the tlb is a pair (logical\_address, physical\_address)

TLB is a circular array, with the oldest element being overwritten once the TLB is full.

tlb tlbArray[TLB\_SIZE];

//The pagetable structure

Declare a char[] 'pagetable' array of size PAGES

pagetable[logical\_page] will be the physical page number for logical page. Value is -1 if that logical page isn't yet in the table.

Char pagetable[PAGES];

//The main\_memory structure

Declare a char[] 'main\_memory' array of size MEMORY\_SIZE

Char main\_memory[MEMORY\_SIZE];

//Searching the TLB

int search\_tlb(unsigned char logical\_page){

//searches the tlb structure for a pair whose first element

//is logical\_page

//If found, return the corresponding physical\_page

//If not, return -1

}

//Adding to the TLB

void add\_to\_tlb(unsigned char logical, unsigned char physical){

struct tlbentry \*entry = &tlb[tlbindex % TLB\_SIZE];

tlbindex++;

entry->logical = logical;

entry->physical = physical;

}

int main(int argc, const char \*argv[]){

//Read in the Backing Store (Refer to the enclosed C file)

//Open addresses.txt for reading (Refer to the enclosed C file)

//Initialize the pagetable array at -1 in each cell

while(more addresses in addresses.txt){

//Extract offset from the address (Refer to the enclosed C file)

//Extract logical\_page from the address (Refer to the enclosed C file)

//Call search\_tlb() to see if logical\_page is in the TLB

if(found in TLB){

tlb\_hits++;

} else{

//see if logical\_page is in the page table

//i.e., if pagetable[logical\_page] is not -1 if(not found in pagetable){ page\_faults++;

//We need to get the page from the backing store

// Copy page from backing file into physical memory

memcpy(main\_memory + physical\_page \* PAGE\_SIZE, backing + logical\_page \* PAGE\_SIZE, PAGE\_SIZE);

//Record this new mapping in pagetable pagetable[logical\_page] = physical\_page;

}

//Add (logical\_page, physical\_page) to the TLB

}

//Print this (logical\_address, physical\_address) pair

}

//print the stats

//number of translated addresses

//number of page faults

//Page fault rate

//Number of tlb hits

//TLB hit rate

return 0;

}