April 20, 2017
Marbo Cheng
CS 326: Operating System

Program 3: Virtual Memory Manager

Files: program3.c, addresses.txt, BACKING_STORE.bin

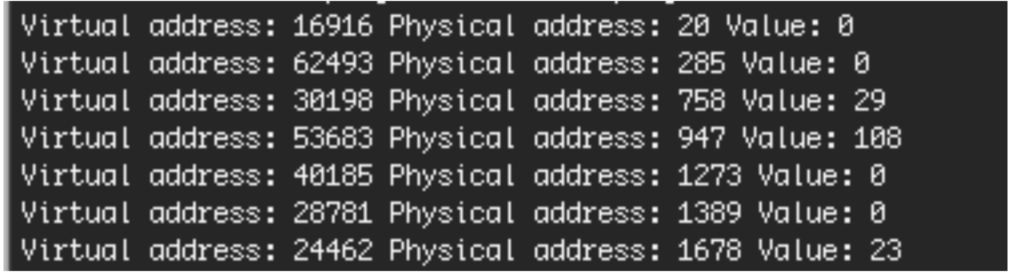
Compile: gcc –o program3 program3.c
Run: ./program3 addresses.txt

This program act as a virtual memory manager which translates logical addresses to physical address for a virtual address space of size 2^16 = 65,536 bytes. The program reads the addresses.txt file, which contains logical addresses. It will use a TLB and page table to translate each logical address to a physical address and output the value for the byte stored at the translated physical address.

The program contains three functions which helps us simulate a virtual memory manager:TLBinsert,storeRetrieve, and retrievePage. TLBinsert(int pageNumber, int frameNumber) uses page number and frame number, checks to see if page number is in TLB if not insert into TLB if it has space otherwise use First In First Out replacement. storeRetrieve(int pageNumber) uses fseek and use SEEK_SET to look from beginning of file to find where data is stored and uses fread to read data into the buffer and put into physical memory. retrievePage(int logical_address) takes in logical address, calculates page number by using addressMask (logical_address & 0xFFFF) and shifting 8 bits to right, calculates offset by using offsetMask (logical_address & 0xFF), looks for match in TLB if TLB index equals page number we get frame number and counter of TLBhits is increased, if frameNumber is not found we go through page table if found in page we get frame number, if frameNumber is not found then we call storeRetrieve(pageNumber) to fetch from BACKING_STORE.bin, increase page fault count, and set framenumber to current firstAvailableFrame, call TLBinsert to insert page number and page frame into TLB, we get value from physical memory, and print virtual address, physical address and value.

When we execute the program main runs calls fopen() on BACKING_STORE.bin and addresses.txt which is provided on command line, it calls retrievePage on each logical address which start the simulation of virtual memory manager, then we calculate page fault rate and TLB hit rate with TLB hits count and Pages Fault count provided calculated on retrievePage which we then divide by total number of translated logical address to calculate the rate, and lastly we safely close the files.

Figure 1. Output of program3.c



```
Virtual address: 16916 Physical address: 20 Value: 0
Virtual address: 62493 Physical address: 285 Value: 0
Virtual address: 30198 Physical address: 758 Value: 29
Virtual address: 53683 Physical address: 947 Value: 108
Virtual address: 40185 Physical address: 1273 Value: 0
Virtual address: 28781 Physical address: 1389 Value: 0
Virtual address: 24462 Physical address: 1678 Value: 23
```