Lab Report #3: Virtual Memory Manager

Sarem Shalforoosh

Linh Ngo

CSC 345-01: Operating Systems

The College of New Jersey

Date of Submission: 4/9/2021

Table of Contents

Project Overview - Virtual Memory Manager

The students will be implementing a Virtual Memory Manager that will translate logical addresses into physical addresses for a virtual address space of size 5,536 bytes. There will be 2 source code files containing two versions of the Virtual Memory Manager. Both versions will read a file containing logical addresses and will use a TLB and page table to translate those addresses into physical addresses. However, one of the source codes will assume that physical memory is the same size as the virtual address space. The other source code will implement a page replacement policy using FIFO to resolve page faults when there is no free memory. The last part of the report will compare the absolute number of page faults and the page fault rate when the physical address space changes.

Specifics

This program initially reads in a file that contains 32-bit integers that represent logical addresses. The program will mask the rightmost 16 bits of the address, because this project is only concerned with 16-bit addresses. The 16 bits are divided into an 8-bit page number and an 8-bit page offset. Other specifics the program supports include:

• 28 entries in the page table

• Page size of 28 bytes

• 16 entries in the TLB

• Frame size of 28 bytes

• 256 frames

• Physical memory of 65,536 bytes (256 frames × 256-byte frame size)

Address Translation

The main requirement of this program is address translation through the use of a TLB and page table. The program translates logical addresses into physical addresses. After a page number is extracted from the logical address, the program searches for the page number in the TLB. If the page number is found, it is a TLB hit. If the page number is not found in the TLB, then the page table is consulted for the page frame and this is considered a TLB miss. If the frame number is not found in the page table, then a page fault occurs.

Handling Page Faults

In the case of page faults, the program will read in a 256-byte page from the provided file BACKING_STORE and store it in an available page frame in physical memory. Then, the page will be stored in the TLB and page table.

Running the Program

To run this program, the user would execute the program in the command line as so:

```
$ ./main addresses.txt
$ ./main_pr addresses.txt
```

The program will read in the file addresses.txt, which contains an arbitrary number of logical addresses ranging from 0 to 65535. Then, the translated logical addresses will be outputted in a file named out1.txt. The corresponding physical address will be outputted in a file named out2.txt. The signed byte value stored in physical memory at the translated physical address will be outputted in a file named out3.txt.

Results & Statistics

Without Page Replacement

Page faults = 244 / 1000, 0.24

TLB hits = 54 / 1000, 0.054

With Page Replacement, Using 128 Frames

Page faults = 538 / 1000, 0.54

TLB hits = 54 / 1000, 0.054

With Page Replacement, Varying Number of Frames

Main.c:

Number of Frames:	32	64	128	256	512
Absolute # of Page Faults:	244	760	538	538	538
Page Fault Rate:	0.24	0.76	0.54	0.54	0.54

Main_pr.c:

Number of Frames:	32	64	128	256	512
Absolute # of Page Faults:	881	760	538	244	244
Page Fault Rate:	0.88	0.76	0.54	0.24	0.24

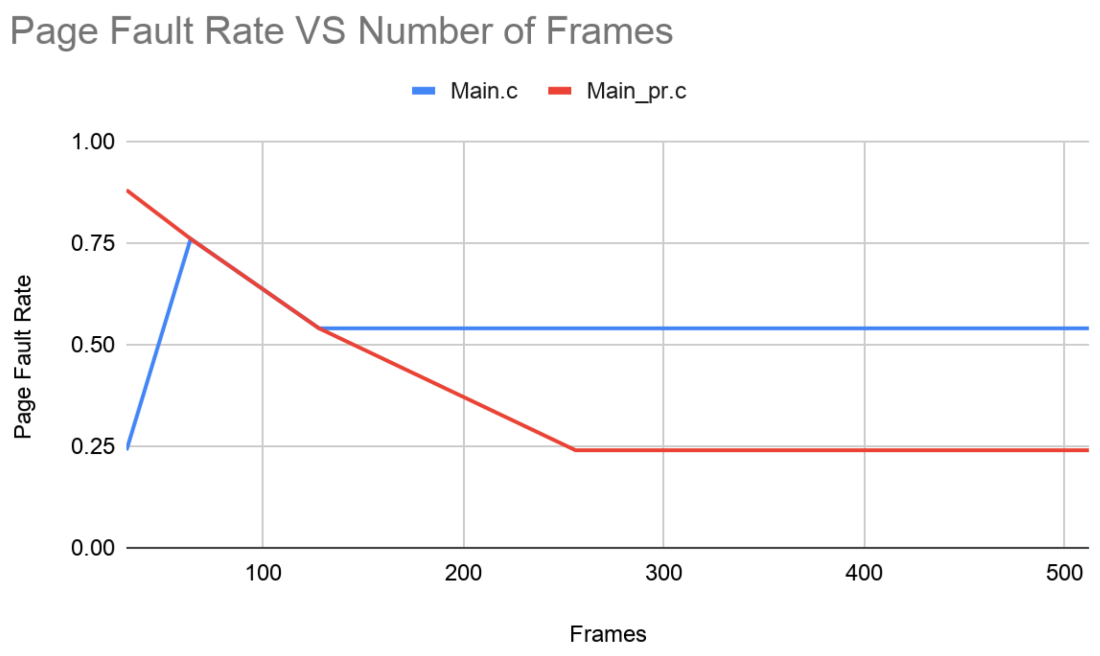

Figure 1: Graph of Page Fault Rate VS Number of Frames

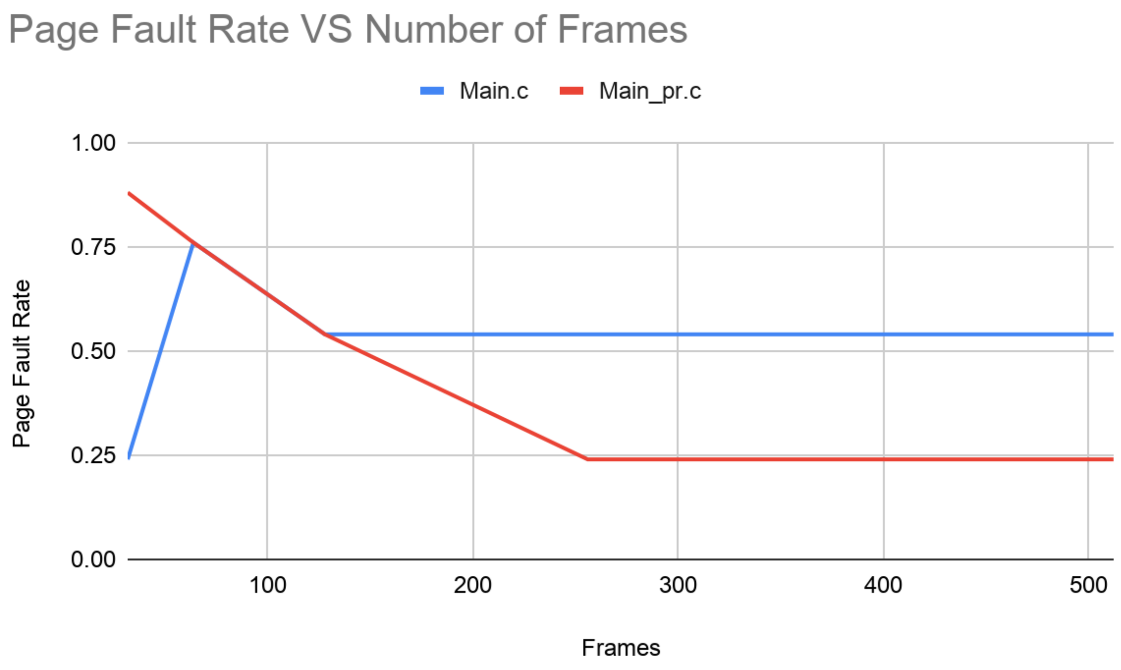

Figure 1: Graph of Number of Page Fault VS Number of Frames

From the results, we can conclude that for the code with page replacement, the page fault rate improves (decreases) when the number of frames increases. This page fault rate, however, plateaus when the number of frames reaches 256. For the code without page replacement, the lowest page fault rate is when the number of frames equals 32. Then the page fault rate steeply increases when the number of frames reaches 64. However, the page fault rate decreases at 64 frames, and plateaus when the number of frames equals 128.