Lab #2 - Virtual Memory Simulator

M. Rajiullah

Karlstad University — September 24, 2024

Preparations

- Read Section 9.2, "Demand Paging", in the 9th edition of the textbook, or Section 10.2, "Demand Paging", in the 10th edition of the textbook.
- Read Section 9.4, "Page Replacement", in the 9th edition of the textbook, or Section 10.4, "Page Replacement", in the 10th edition of the textbook.

Description

This lab aims to help the students deepen their understanding of virtual memory management and demand paging. The students are asked to implement a virtual memory simulator, `vmsim`, that simulates the behavior of three page replacement algorithms, FIFO, Optimal, and LRU, when pure demand paging is used. The simulator has three options, the employed page replacement algorithm (`-a`), the number of frames in the physical memory (`-n`), and the name of the trace file with (virtual) memory addresses from a fictive process (`-f`). The synopsis for the command is as follows:

```
vmsim -a <fifo | optimal | lru> -n <number of frames> -f <trace file>
```

When invoked, `vmsim`, runs through the (virtual) memory addresses in `<trace file>` and simulates the actions taken in terms of accesses to the page table, the physical memory, and the backing store, when a fictive process wants to read from those addresses. For each memory access, the simulator should display: the memory address accessed, whether the accessed page is in the page table (page table hit) or if it needs to be retrieved from the backing store (page fault), and, finally, if the memory access renders a page replacement (page out/page in). When the simulation ends, a summary should be printed out: the size of the physical memory in number of frames, the total number of memory accesses, the total number of page hits, the number of page faults, and the total number of page replacements.

The size of the virtual memory is 64 KB and is addressed with 16-bit addresses (0x0000 - 0xFFFF); the frame size is 256 bytes, i.e., the size of the physical memory is `<frames>` × 256 bytes. In other words, if the number of frames is 256 then the physical memory is 64 KB, and no page replacements need to take place. Setting the number of frames to 0 should not be permitted.

The trace file with the virtual memory addresses, whose access are to be simulated by `vmsim`, has the following format:

```
<memory address in hexadecimal> <newline character>
```

A sample trace file, `trace.dat`, is shown below. Sample output for the three possible page replacement algorithms is shown in Figure 1.

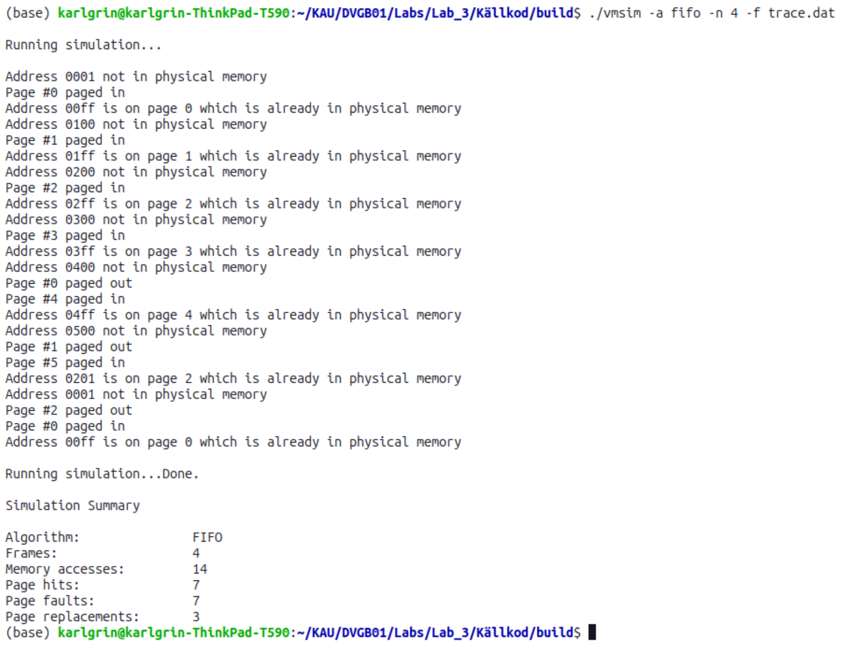
File

trace.dat

```
0x0001
0x00FF
0x0100
0x01FF
0x0200
0x02FF
0x0300
0x03FF
0x0400
0x04FF
0x0500
0x0201
0x0001
0x00FF
```

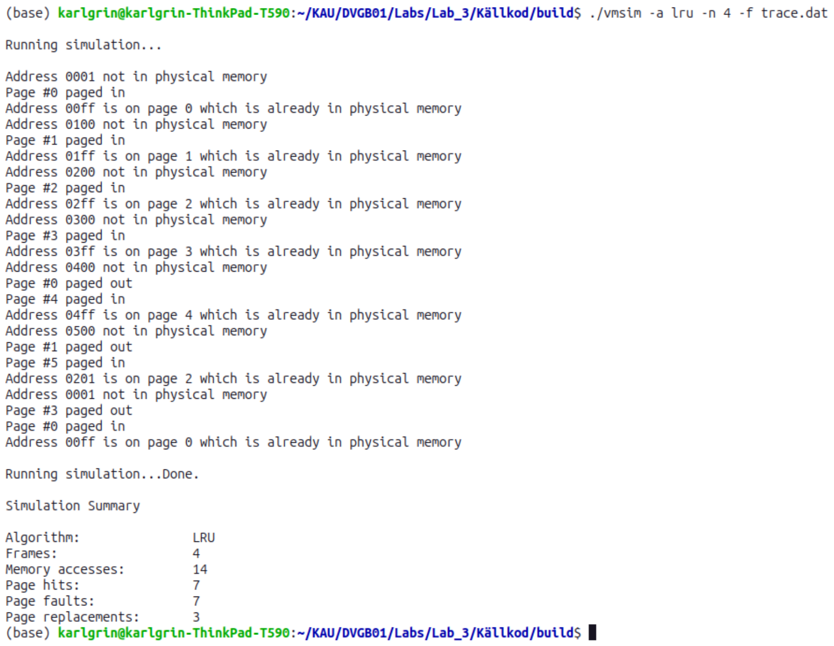
Examination

The lab is graded as *pass* or *failed*. To pass, the students should demonstrate their virtual memory simulator to a lab assistant.



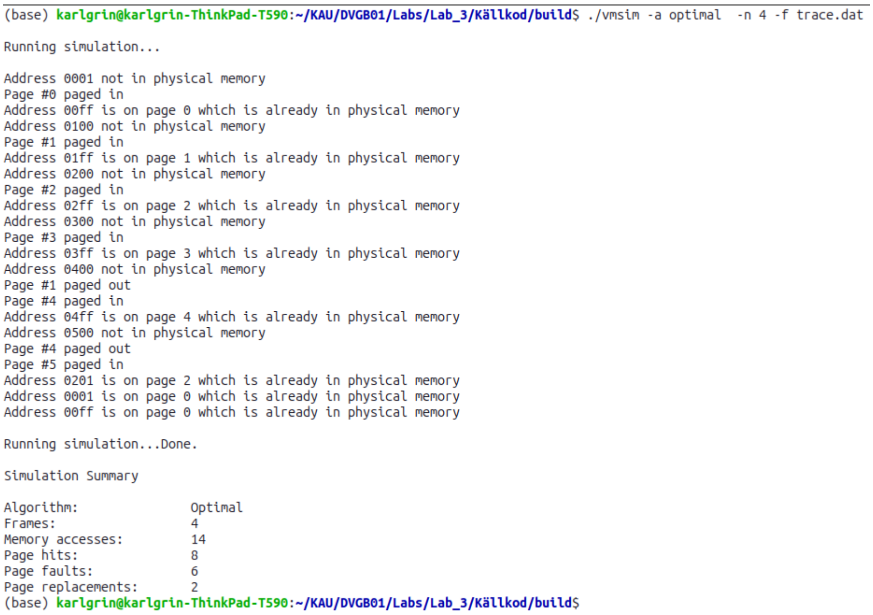
```
(base) karlgrin@karlgrin-ThinkPad-T590:~/KAU/DVGB01/Labs/Lab_3/Källkod/build$ ./vmsim -a fifo -n 4 -f trace.dat

Running simulation...

Address 0001 not in physical memory
Page #0 paged in
Address 00ff is on page 0 which is already in physical memory
Address 0100 not in physical memory
Page #1 paged in
Address 01ff is on page 1 which is already in physical memory
Address 0200 not in physical memory
Page #2 paged in
Address 02ff is on page 2 which is already in physical memory
Address 0300 not in physical memory
Page #3 paged in
Address 03ff is on page 3 which is already in physical memory
Address 0400 not in physical memory
Page #0 paged out
Page #4 paged in
Address 04ff is on page 4 which is already in physical memory
Address 0500 not in physical memory
Page #1 paged out
Page #5 paged in
Address 0201 is on page 2 which is already in physical memory
Address 0001 not in physical memory
Page #2 paged out
Page #0 paged in
Address 00ff is on page 0 which is already in physical memory

Running simulation...Done.

Simulation Summary

Algorithm:               FIFO
Frames:                  4
Memory accesses:         14
Page hits:               7
Page faults:             7
Page replacements:       3
(base) karlgrin@karlgrin-ThinkPad-T590:~/KAU/DVGB01/Labs/Lab_3/Källkod/build$
```

(a) FIFO



```
(base) karlgrin@karlgrin-ThinkPad-T590:~/KAU/DVGB01/Labs/Lab_3/Källkod/build$ ./vmsim -a lru -n 4 -f trace.dat

Running simulation...

Address 0001 not in physical memory
Page #0 paged in
Address 00ff is on page 0 which is already in physical memory
Address 0100 not in physical memory
Page #1 paged in
Address 01ff is on page 1 which is already in physical memory
Address 0200 not in physical memory
Page #2 paged in
Address 02ff is on page 2 which is already in physical memory
Address 0300 not in physical memory
Page #3 paged in
Address 03ff is on page 3 which is already in physical memory
Address 0400 not in physical memory
Page #0 paged out
Page #4 paged in
Address 04ff is on page 4 which is already in physical memory
Address 0500 not in physical memory
Page #1 paged out
Page #5 paged in
Address 0201 is on page 2 which is already in physical memory
Address 0001 not in physical memory
Page #3 paged out
Page #0 paged in
Address 00ff is on page 0 which is already in physical memory

Running simulation...Done.

Simulation Summary

Algorithm:               LRU
Frames:                  4
Memory accesses:         14
Page hits:               7
Page faults:             7
Page replacements:       3
(base) karlgrin@karlgrin-ThinkPad-T590:~/KAU/DVGB01/Labs/Lab_3/Källkod/build$
```

(b) LRU



```
(base) karlgrin@karlgrin-ThinkPad-T590:~/KAU/DVGB01/Labs/Lab_3/Källkod/build$ ./vmsim -a optimal  -n 4 -f trace.dat

Running simulation...

Address 0001 not in physical memory
Page #0 paged in
Address 00ff is on page 0 which is already in physical memory
Address 0100 not in physical memory
Page #1 paged in
Address 01ff is on page 1 which is already in physical memory
Address 0200 not in physical memory
Page #2 paged in
Address 02ff is on page 2 which is already in physical memory
Address 0300 not in physical memory
Page #3 paged in
Address 03ff is on page 3 which is already in physical memory
Address 0400 not in physical memory
Page #1 paged out
Page #4 paged in
Address 04ff is on page 4 which is already in physical memory
Address 0500 not in physical memory
Page #4 paged out
Page #5 paged in
Address 0201 is on page 2 which is already in physical memory
Address 0001 is on page 0 which is already in physical memory
Address 00ff is on page 0 which is already in physical memory

Running simulation...Done.

Simulation Summary

Algorithm:               Optimal
Frames:                  4
Memory accesses:         14
Page hits:               8
Page faults:             6
Page replacements:       2
(base) karlgrin@karlgrin-ThinkPad-T590:~/KAU/DVGB01/Labs/Lab_3/Källkod/build$
```

(c) Optimal

Figure 1: Screenshots from the execution of `vmsim` with the sample trace file: `trace.dat`.

End of Lab