

Designing a Virtual Memory Manager

Please refer to the Programming Projects on page 458-461 of the textbook for the description of this Programming Assignment on Designing a Virtual Memory Manager.

As mentioned in the textbook, the assignment includes two parts. For the second part (Modifications), please modify your program developed in part 1 to implement two strategies for page-replacement in physical memory. They are FIFO (First in and First out) and LRU (Least Recently Used).

The following files and sample program can be downloaded from canvas:

BACKING_STORE.bin
part1.out address.txt
correct.txt part2.out
addresses1.txt
correct1_fifo.txt
correct1_lru.txt
addresses2.txt
correct2_fifo.txt
correct2_lru.txt

Specification:

Part 1:

Your program should run as follows:

```
./part1.out BACKING_STORE.bin addresses.txt
```

This should produce an output file by name “correct.txt”, which shows the address translation of each address. Please look through the provided ‘correct.txt’ file for the specific format. (Please use ./part1.out, address.txt, and correct.txt provided to verify this part of assignment)

Part2:

Due to the additional constraints added, for part2, your program should run as follows.

```
./part2.out BACKING_STORE.bin addresses.txt strategy
```

Strategy can be “fifo” (First In First Out) or “lru” (Least Recently Used) .

As above, the name of the output file should be correct.txt, and has the same output format as part1. (To verify your program, please use part2.out, address1.txt, correct1_fifo.txt, correct1_lru.txt, address2.txt, correct2_fifo.txt, correct2_lru.txt).

Submission:

The assignment will be submitted in two parts. Part1 would contain the source files and makefile that implement Part 1 of the assignment, and Part2 for the second part. Both are kept in separate folders by name 'part1' and 'part2' respectively.

Like any other programming assignment of this course, your program must be verified to run on cse.unl.edu.

Include a README file in with your code. Use the README file to:

- Document instructions to compile, run, and test your program.
- Document how you have tested your program, report on the testing results and what the testing results demonstrate.
- Document any problems you had.
- Answer to the following “fun” question.

The readme file can be placed outside folders part1 and part2.

Use webhandin to handin your assignment. Submit a single zip file, <cse-accountname>_pa4.zip containing all the files mentioned above.

Due Date: Friday April 27th at 11:59PM

Question:

In this assignment, we assume there is only one process running in the system and all the memory belongs to it. However, in a typical computer system there will be multiple processes at any given time. Thus, the physical memory will be shared among all the processes. Briefly discuss (in no more than 150 words) the challenges that you might encounter when extending our model to a multiprogramming system.

Note: Think about how you would distribute the physical memory among all the processes, and the strategies you could use to do so. How do these strategies compare with each other? What if some process has heavy memory requirement compared to others, or if it needs a lot of memory in bursts? You can focus on one particular strategy or briefly touch upon all of them.

We are not expecting a bookish answer (hence “fun” question). So, feel free to go bonkers, think, speculate and come up with your own awesome strategies. Ladies and Gentleman, the floor is yours.