UNIVERSITY OF WISCONSIN
Computer Sciences Department

CS 537
Fall 2020

Barton Miller

Programming Assignment #5 (Extra credit): Page Replacement Simulator

Handed out: Friday, November 16
Due: Friday, December 11, 5pm

Extending the Replacement Simulator with MIN/OPT

The goal of this assignment is to understand how to implement a simulator that has an optimal scheduling algorith that uses future knowledge. The basis for this assignment is your implementation for Programming Assignment #4.

Note that you must start with your own implementation of Program 4, with no reference to anyone else's. Of course, this means that your code for Program 4 must be working well.

Note that no late days can be used for this assignment.

A. Running your Program

You will build a 4th executable called "537pfsim-opt". All other structural details of the simulator will be the same as for Program 4.

B. The Trace Files

You will use the same trace files as for Program 4.

The trace files can be found on AFS at `~cs537-1/public/proj4`.

C. Program Information

You will add a fourth page replacement algorithm to your simulator; only the page replacement algorithm module should change. This means that only the files associated the the replacement algorithm should change. All other details will be the same as for Program 4.

C.3. Scheduling Algorithms

You will implement the OPT/MIN page fault replacement algorithm. The basis for this algorithm is that it will select the page in memory that will not be referenced until furthest in the future. Of course, this requires future knowlege, which requires you to scan ahead in trace file. Doing this efficiently will be a definite challenge.

By "furthest in the future", we mean the memory reference that is latest in the trace file.

This is a global replacement algorithm.

C.4. Simulator Parameters

Your simulator parameters will stay the same as for Program 4.

C.5. Performance Data

Your simulator will keep track of the same statistics as for Program 4.

D. Software Design Issues

Good design on this assignment will save you, literally thousands of lines of code. The page replacement algorithm should be encapsulated in a PageAlgorithm module. In one version of the program, this module will do something simple, like a FIFO scan of pages. In another version, it may need much more complicated bookkeeping information to track last reference times for LRU.

All other parts of your program should be the same, so you can re-use them for the different versions.

Some of the modules that you might want to build will be for input, processes, page tables, page frame table, paging device (disk queue), and statistics.

Your makefile will have build rules for the four separate programs: `537pfsim-fifo`, `537pfsim-lru`, `537pfsim-clock`, and `537pfsim-opt`.

E. Deliverables

You can work individually or in a group of two. In either case, you will turn in a single copy of your program, clearly labeled with the name and logins of both authors.

You will turn in your programs, including **all** .c and .h files and your makefile. Also include a README file which describes a little bit about what you did for this project.

Note that you must run your programs on the Linux systems provided by the CS Department. You can access these machines from the labs on the first floor of the Computer Sciences Building or from anywhere on the Internet using the ssh remote login facility. If you do not have ssh on your Windows machine, you can download this client:

> `SSHSecureShellClient-3.2.9.exe`

Your program should run on the test trace files we provide. These files can be found in `~cs537-1/public/proj4`

You should run your simulator with page sizes of 512 and 4096 bytes and with physical memory sizes of 1 MB, 4MB, and 32 MB. That's a total of 6 runs per trace file per algorithm. (Of course, you'll probably want to use a shell script to run all these different variations.)

F. Handing in Your Assignment

Your CS537 handin directory is `~cs537-1/handin/`*`your_login`* where *`your_login`* is your CS login. Inside of that directory, you need to create a `proj5` subdirectory (unless the TAs created one for you already).

Copying your files to this directory is accomplished with the `cp` program, as follows:

```
    shell% cp *.[ch] makefile README ~cs537-1/handin/your_login/proj5
```

Make sure to hand in **all** the files for your program, including the files that implemented the algorithms for Program 4.

You can hand files in multiple times, and later submissions will overwrite your previous ones. To check that your files have been handed in properly, you should list `~cs537-1/handin/`*`your_login`*`/proj5` and make sure that your files are there. The handin directories will be closed after the project is due.

Whether you are working individually or in pairs, you should:

1. Submit only one copy of your code.
2. Create a file called partner.txt in each of of your `proj5` directories. It should have a line in the file for each person who worked on the code (so, 1 or 2 lines). Each line will have you name, CS login and NetID.

G. Original Work

This assignment must be the original work of you and your project partner. Unless you have explicit permission from Bart, you may not include code from any other source or have anyone else write code for you.

Use of unattributed code is considered plagiarism and will result in academic misconduct proceedings (and "F" in the course and a notation on your transcript).

Last modified: Mon Nov 16 13:24:03 CST 2020 by bart