Name: Adrian Galicia and Ivan Acosta
username: masc1442 and masc1461
Project: CS570 Assignment 2
File: README

File manifest (name of all files included in this project):

a2.cpp
Makefile
a2.h
pages.txt
pages2.txt
a2
README

Compile Instructions:

Type make in the a2 directory.

Operating instructions (include any/all command line options/arguments, required inputs, etc):

To run the program just run the executable by typing ./a2. The sequence of pages the page/thread accesses is in pages.txt as required in the instructions. To use a pages file with different data in it you can modify the a2.h file where it says char* file_name = “pages.txt” and change it to pages2.txt instead of pages.txt,then run make again and run the executable by typing ./a2.

Below is an additional option to run the program that is included in a2.cpp at the top of the file under usage.

./a2 -f <file name> -n <number_of_frames> -l <line to be read(1-n)>, if -l is not used program will read the first line only

List/description of novel/significant design decisions:

N/A.

List/description of any extra features/algorithms/functionality you included which were not required:

Also added the FIFO algorithm in addition to the required LRU,Clock and OPT algorithms.

List/description of all known deficiencies or bugs:

No known deficiencies or bugs.

Lessons Learned:

The OPT algorithm definitely earns the right to be called optimal.

LRU and FIFO algorithms differ only in one aspect. LRU brings an element to the front if it is already in memory. FIFO doesn't do anything if the element is already in memory. if a true value is passed in the argument it behaves as FIFO otherwise it behaves as LRU.

Team Project

This project was done as a team. Please use account masc1442 for grading.

Analysis

We ran some tests using two different pages files. One containing one line of text and one containing 35 lines of texts with more varied numbers. The first test we entered three different numbers of frames. The second test we tried six different numbers of frames. Below are the results from the terminal.

First Test

```
> ./a2

please enter the number of frames 3

number of frames is: 3, reading file: pages.txt

running OPT .....page_faults 6

running LRU .....page faults = 7

running FIFO.....page faults = 9

running CLOCK.....page faults = 8

> ./a2

please enter the number of frames 4

number of frames is: 4, reading file: pages.txt

running OPT .....page_faults 5

running LRU .....page faults = 6
running FIFO.....page faults = 6
running CLOCK.....page faults = 6
> ./a2
please enter the number of frames 5
number of frames is: 5, reading file: pages.txt
running OPT .....page_faults 5
running LRU .....page faults = 5
running FIFO.....page faults = 5
running CLOCK.....page faults = 5
```

Second Test

We also ran tests with a pages file called pages2.txt that had 35 lines of text. This one produced vastly different results from the tests with pages.txt. Below are the results.

```
> ./a2
please enter the number of frames 5
number of frames is: 5, reading file: pages2.txt
running OPT .....page_faults 540
running LRU .....page faults = 802
running FIFO.....page faults = 805
running CLOCK.....page faults = 802
> ./a2
please enter the number of frames 7
number of frames is: 7, reading file: pages2.txt
running OPT .....page_faults 435
running LRU .....page faults = 718
running FIFO.....page faults = 733
running CLOCK.....page faults = 728
> ./a2
please enter the number of frames 30
number of frames is: 30, reading file: pages2.txt
running OPT .....page_faults 40
running LRU .....page faults = 48
running FIFO.....page faults = 89
running CLOCK.....page faults = 59
> ./a2
please enter the number of frames 25
number of frames is: 25, reading file: pages2.txt
running OPT .....page_faults 60
running LRU .....page faults = 105
running FIFO.....page faults = 158
running CLOCK.....page faults = 111
> ./a2
please enter the number of frames 40
number of frames is: 40, reading file: pages2.txt
running OPT .....page_faults 38
running LRU .....page faults = 38
running FIFO.....page faults = 38
running CLOCK.....page faults = 38
> ./a2
please enter the number of frames 50
number of frames is: 50, reading file: pages2.txt
```

```
running OPT .....page_faults 38

running LRU .....page faults = 38

running FIFO.....page faults = 38

running CLOCK.....page faults = 38
```

The results from using pages2.txt produce much different results than when using pages.txt. The first two tests using a small number of frames show that LRU,FIFO and Clock produce almost the same amount of page faults while OPT works much better and produces a much smaller number of page faults.

The third test we entered a 30 frames and this still showed that OPT was still the most efficient algorithm but the number of page faults was much closer to LRU and CLOCK while LRU and clock were only separated by 11 page faults with CLOCK having more. FIFO ended up producing the most page faults of all the algorithms in this test.

The fourth test with 25 frames varied more from the previous test. OPT remains the best,while LRU and CLOCK are very close in the number of page faults they produce. FIFO remained the one that produced the most page faults in this test as well,producing 47 more than the second worst algorithm.

The fifth and sixth tests used 40 and 50 frames and at this point all the algorithms performed the same and produced the same amount of page faults which was 38. What we noticed with this was after reaching a certain number of frames the algorithms all performed the same and produced just 38 page faults. We also tested the algorithms using the pages2.txt file with 99 frames and 2000 frames and the same number of page faults was produced. Results are below.

```
please enter the number of frames 99

number of frames is: 99, reading file: pages2.txt

running OPT .....page_faults 38

running LRU .....page faults = 38

running FIFO.....page faults = 38

running CLOCK.....page faults = 38

> ./a2

please enter the number of frames 2000

number of frames is: 2000, reading file: pages2.txt
```

```
running OPT .....page_faults 38

running LRU .....page faults = 38

running FIFO.....page faults = 38

running CLOCK.....page faults = 38
```

Conclusion

Using a pages.txt file with a single line of text as instructed shows all the algorithms produce a similar number of page faults but OPT is always the one with the least page faults. Once we enter 5 frames or more all the algorithms produce 5 page faults.

When using pages2.txt with 35 lines of text and more variance in the numbers included in the file the results are more varied. Most of the time LRU and CLOCK were not that far off from each other with their number of page faults and with a small number of frames FIFO wasn't that far off from LRU and CLOCK. OPT was always the best though. With 30 and 25 frames however FIFO's results were worse than LRU and CLOCK while OPT still remained the best.

The conclusion appears to be that using a pages file with a single line of text does not show a vast difference in the number of page faults each algorithm produces and when entering 5 or more frames they all produce the same amount of page faults.

However, using a pages file with many more lines of text inside it and more varied numbers produces much more varied results until a certain point where a certain number of frames is entered and past that point they all produce the same amount of page faults,as shown with the tests using 40,50,99 and 2000 frames.