//================================================================

// Name : README

// Author : Katie Brigham MASC1420, Shawn Vega MASC1440

// Description: Assignment 2, CS570 Summer 2014

//================================================================

File manifest:

a2.cpp

a2.h

makefile

README

Compile instructions:

Input “make”

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

After compiling with “make”, input “a2” to initiate the executable

Program will then prompt for number of frames. Enter number and press “Enter”

List/description of novel/significant design decisions:

Studying these algorithms, one of the most important decisions was to identify

The many common steps that were called, and create helper methods to increase

efficiency. Each algorithm then was created using helper methods.

List/description of any extra features/algorithms/functionality:

Error checking for page number within requirements.

Prompts user for page number.

FIFO algorithm

Random Poisson distribution with a lambda of 5

**List/description of all known:** Random numbers were generated from a random poisson distribution

with a lambda of 5. The range of numbers generated were integers from 0-12, causing the higher number

of frames selected to produce a plateau of 12 faults. This meant that to use our test sets of generated

random numbers we were limited to lower frame numbers, therefore our analytical conclusions are

bases solely on patterns observed with these frame numbers.

**Analysis of the page replacement algorithms:**

**-Control**: Program algorithms

**-Variables/input characteristics:** Test input was generated by Poisson distribution

We tested input data sets 1 & 2, each with all 3 frame sizes.

-**Input data set 1**: 6 4 7 4 5 6 3 6 4 6 3 4 7 2 5 3 11 2 6 4 1 7 7 4 2 6 8 3 9

7 3 3 7 5 4 6 9 5 2 5 5 6 7 7 5 4 5 5 6 7 9 6 5 3 1 3 6 8 6 5 6 3 6 5 5 5 6 4

9 6 4 6 10 4 2 6 7 4 13 6 5 5 6 6 1 4 6 3 4 6 3 8 2 5 2 6 4 6 3 6 6 4 2 4 5 4

6 7 2 7 8 7 5 5 5 3 6 6 4 10 2 6 5 4 5 5 3 8 5 8 10 1 4 8 2 0 2 9 8 5 4 3 6 5

5 7 7 4 0 2 4 6 4 6 13 4 4 6 6 4 7 5 5 3 8 2 1 3 5 4 8 4 7 6 6 4 5 4 4 7 7 5

4 5 7 8 4 4 3 5 8 8 5 7 3 5 2 8 7 4 7

-**Input data set 2**: 4 10 5 10 3 8 5 7 7 3 3 10 3 3 4 7 9 1 7 4 8 7 5 5 8 6 3 6

3 6 3 5 4 3 2 4 7 2 5 7 4 9 7 2 4 4 3 5 5 7 7 9 3 6 5 10 10 2 2 7 5 8 3 5 5 5

4 4 2 5 2 4 4 4 5 3 4 2 6 2 4 8 5 2 6 3 4 7 6 6 3 3 5 2 7 3 8 3 5 7 4 7 6 8 2

5 5 9 6 5 2 5 7 2 3 9 5 3 3 5 4 3 3 4 1 9 4 4 5 5 8 5 2 4 6 5 2 8 6 1 6 3 6 8

4 4 11 9 4 1 3 2 8 8 9 5 3 4 3 3 3 5 2 2 4 7 7 3 4 4 7 4 6 6 3 2 7 1 2 3 6 3

7 2 3 5 3 4 6 7 4 5 5 5 7 5 8 8 3 5 10 2 5 7 4 0 5 6 6 3 5 8 2 5 8 4 9 6 9 8

5 3 7 6 4 4 4 6 7 4 7 4 6 4 5 6 2 6 6 5 5 3 4 2 3 3 5 4 6 3 4 7 4 11 6 3 6 4

7 6 2 3 4 5 3 2 4 4 5 4 4 7 2 8 5 9 8 4 4 6 5 4 6 6 3 6 14 1 7 4 5 6 5 4 8 8

4 4 5

**FRAMES:**

-Frame 1:3

-Frame 2:8

-Frame 3:12

**Data Set 1 Trials:**

-Frame 1:3

OPT: 86 LRU: 118 CLK: 128 FIFO: 130

-Frame 2:8

OPT: 20 LRU: 40 CLK: 41 FIFO: 43

-Frame 3:12

OPT: 12 LRU: 13 CLK: 15 FIFO: 19

-Frame 3:12

**Data Set 2 Trials:**

-Frame 1:3

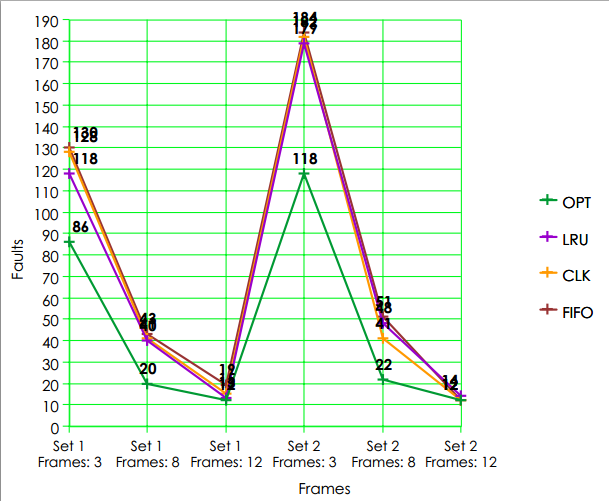
OPT: 86 LRU: 118 CLK: 128 FIFO: 130

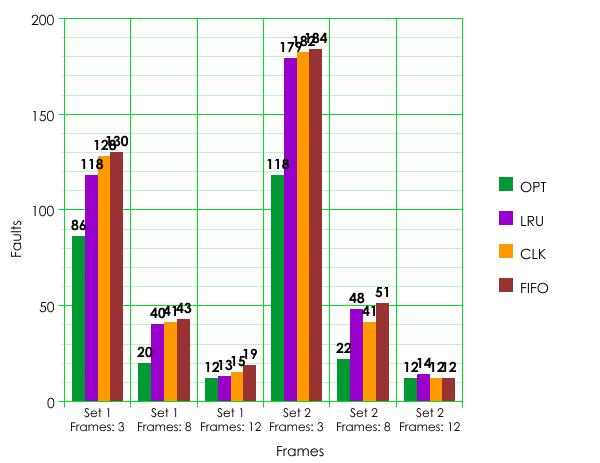
-Frame 2:8

OPT: 20 LRU: 40 CLK: 41 FIFO: 43

-Frame 3:12

OPT: 12 LRU: 13 CLK: 15 FIFO: 19





Conclusion: We conclude that the faults discovered in OPT generally we below the average of faults for the remaining algorithms, exempting the faults counted with the higher frame number of 12. We chose 12 in order to demonstrate how the number of faults will become more static a variability of the numbers lower and the number of frames increase. The algorithm that produced the most faults was the FIFO algorithm, followed by the CLK algorithm then the LRU algorithm.