Programming Assignment 3

Start Assignment

Due Jul 29 by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip **Available** until Aug 2 at 11:59pm

Assignment Objective

Compare the performance of page replacement algorithms for fixed numbers of frames: optimal, FIFO, LRU, second chance.

Description

The simulation generates reference strings based on typical characteristics of program execution. The strings are then used to compare the performance of different algorithms in terms of the page faults generated.

- The virtual memory of a process consists of P pages, numbered 0 through P −1.
- A reference string RS is a sequence of integers in the range [0 : P − 1]. Each element, p, of RS represents one reference to page p.
- The physical memory of the system is approximated as an integer array M[10], where 10 is the number of frames. Each element M[f] represents one frame and contains the number p of the page currently residing in the frame f.

The main distinguishing property of a reference string RS is the degree of locality. A reference string constructed as a sequence of random numbers distributed uniformly over the range [0 : P - 1] has no locality. Such a string represents a situation where too many processes are running concurrently and the lack of locality results in thrashing.

Most programs display a high degree of locality, which can be modeled as follows:

- Consecutive pages are drawn from a small region within [0 : P - 1], referred to as the current locus of reference, L.
- L is defined by a starting position, s, within [0 : P - e] and size, e.
  In reality, L is not a contiguous region since pages from at least three regions (code, data, and stack) are being accessed. For studying page-replacement algorithms, however, only the size of L is of interest and thus a contiguous locus of reference is adequate.

- The locus L gradually shifts to the right, which emulates sequential execution of the program. The rate of the motion is governed by a constant, m, which determines how many pages are chosen from L before L is shifted to the right.
- At random time points, L moves to a new location, s, within [0 : P - e], chosen at random. The change represents the occasional transition of the program to a new area, resulting from a function call or a branch instruction.

Using the above assumptions, a reference string is generated as follows:

```
select parameters for the new RS:        /* RS is an initially empty file */
   virtual memory size: P                /* A large integer. Ex: with address size
                                            of 32 bits and page size of 4096, P = 2^20 */
   starting location: s                  /* Initial s must be 0 */
   size of L: e                          /* Small integer representing current locus */
   rate of motion: m                     /* Each page in L is typically referenced only a few
                                            hundred times. Ex: m = 200 */
   probability of transition: t          /* A real number close to 0 since transitions are rare */

repeat until |RS|>= 1,000,000 is generated
   select m random numbers in the range [s:s+e)   /* Process is executing within current locus */
   append the numbers to RS                       /* RS is a file that grows with each iteration */
   generate random number r in the range [0:1)
   if (r < t), generate new s                     /* Process transitions to a new location s */
   else increment s by 1 (modulo P)               /* Process remains within current locus */
```

Assignment

- Implement a C program to generate reference strings.
- Generate five sets of reference strings by setting the parameters P, m, e and t to the following five combinations, to mimic various program behaviors (high vs low locality, fast vs slow changing locus).

Scenario #	P	m	e	t
1	2^{20}	20	10	0.001
2	2^{22}	20	10	0.002
3	2^{20}	20	15	0.001
4	2^{22}	50	8	0.001
5	2^{18}	7	15	0.002

- Implement the four page replacement algorithms in C. If your group has less than four members, three algorithms are enough and implementing all four will give you a 20% extra bonus credit. If your group size is 4, then, you need to implement all the four algorithms.
- Assuming that a process will be given exactly 10 frames, record and tabulate the numbers of page faults generated by different algorithms under each of the five aforementioned scenarios.

Deliverables

- You need to submit a zip file containing the following information:
  - A readme.txt file containing
    - List of team members name and panther id.
    - List of items that each member of the group is supposed to do. The list must be agreed by group members during their first synchronized meeting
    - List of items that each member of the group has completed (please prepare this list before submitting your project).
    - Minutes, participants, and agenda of at-least two synchronized meetings of group members (one should happen within the first five days and one within the second five days after assignment available date.
  - Source files of your simulation program
  - A single report.pdf containing the tables showing the number of page faults for different algorithms under different scenarios.