**CSC 345 Operating Systems**

**Class 17 October 8, 2015**

**Virtual Memory Project**

**Due: Tuesday, October 27, 2015, 11:00 p.m.**

**Reminder: Midterm exam is on Thursday, October 15, 2 p.m. in Forcina 409.**

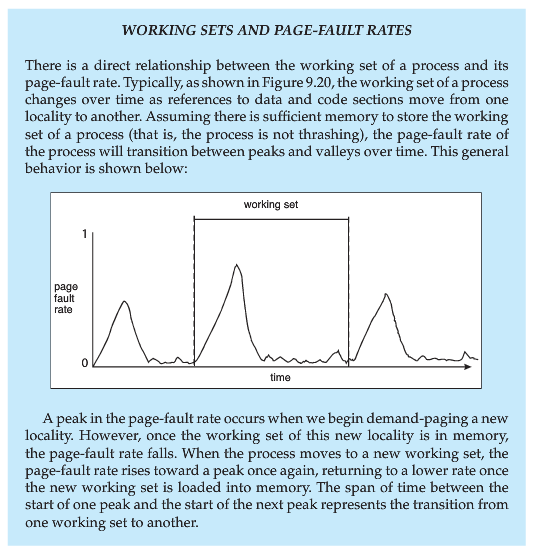
*You may work with one lab partner on this project. Only one submission is to be made; make certain that both of your names are included on ALL components submitted. Each partner must contribute to the coding, the data generation, the analyses, and the report writing. No one can shift those responsibilities to the other partner. If a partner is not holding up their end of the project, please let me know. I will dissolve the partnership and both students must complete the project independently.*

Consider the locality of reference exhibited during execution of a process. The occurrence of temporal locality can be illustrated in a reference string when reuse of pages is depicted, e.g., when the reference string shows repeated references to the same page […4, 4, 4, 4, 4, 4, 5, 12,…]. In this reference string, there is a high degree of temporal locality, and a high probability of reuse. We observe this when executing code in a loop, or when accessing a local variable during each iteration of a loop.

Spatial locality can be depicted in a reference string when the probability of accessing page k+1 is high because page k was addressed. For example, for a reference string of [1,2,3,4,5…], it is highly probable that page 4 is going to be referenced because page 3 was accessed. The observation of spatial locality suggests that a pre-paging algorithm could help improve cache performance. Spatial locality may be observed when referencing the elements of an array in succession, or when sequentially executing a function.

Persistence is the tendency to repeat the use of a page and is similar to temporal locality. Concentration is the tendency of the references to use a small subset of the whole address space. A reference string with a high concentration would exhibit good caching performance ultimately leading to large performance gains in a demand paging system supported by a cache. Persistence is measured by counting the consecutive references to the same address. Concentration is measured by computing the fraction of the address space used for a large fraction of the reference string. [Wu 2007, Packet Forwarding Technologies, CRC Press, pp 80-81.]

Study the following graphic and accompanying discussion from your textbook, Operating Systems Concepts, Silberschatz, page 431.



**Part 1**

Develop a fully documented C program to generate reference strings of a requested length, drawn from pages numbered 1-100. For discussion, please use a reference string length of 250.

1. Develop a program to generate reference strings that simulate spatial locality over small and large regions of memory.
2. Enhance the program to generate reference strings that exhibit temporal locality over small and large periods of time.
3. Further extend the program to produce a reference string that contains the characteristics of both spatial and temporal locality.
4. Finalize the program to generate pseudo-random reference strings.

In your lab report, for types a-d, you must provide a graphic of the reference string and an analysis, explaining why the particular characteristic(s) is(are) simulated in the generated reference string.

**Part 2**

For each type of reference string, you are to describe the optimal page replacement algorithm. Include this discussion for characteristic reference strings a-d.

**Part 3**

Create a fully documented simulation program in C to investigate how the number of pages allocated to a process affects the number of page faults generated. You are to implement this simulation to examine the impact of the following page-replacement strategies of: FIFO, true LRU, and second chance. Note: Since this is a simulation, you are able to implement LRU without depending on an approximation. Your simulation program must produce statistics on the performance of each replacement strategy. You are to include your analysis and discussion in the lab report.

**Part 4 (*This part must be completed satisfactorily for an A level grade*)**

Modify your program from Part 3 to simulate the effect of multiprogramming and global replacement of pages by applying a round-robin scheduler with a time quantum of 10 to the processes. Simulate process admittance by assigning arrival times during setup. Include a simulated system “clock” to determine simulated process arrival. Simulate the job loads by varying the length and types (a-d) of the reference strings. Generate the necessary statistics on performance. Include your analysis and discussion in the lab report.

**Guidelines and expectations**

Code must be formatted and documented appropriately.

Your work is to be conducted on one of the Linux lab machines using the cc compiler.

**Lab report –** Each section must be professional in nature and clearly outline key points.

Abstract

Problem statement (one for each Part)

Methodology (coding solution discussion for each part)

Data (collected through experimentation/simulation)

Analyses

Conclusions

References and/or Resources

Canvas submission of programs (.c files) and report is due on Tuesday, October 27, before 11 p.m.   No extensions.  Late submissions will not receive credit. Programs that are not complete, or do not compile for the instructor, will receive a 0 score. If you do not turn in an appropriate lab report, you will not earn any credit for this project. Turnitin will be enabled.