CS323 – Exercises
Week 7
4 April 2019

**Exercise 1:**

Assume a machine in which 4 page frames are available, and in which memory is initially empty. Consider the following reference string of memory accesses to pages:
1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
How many page faults will occur with the following replacement algorithms? Do not count the initial 4 page faults to bring pages 1 to 4 in memory. Briefly explain your answer.

a) OPT
b) LRU
c) FIFO
d) CLOCK

**Answer:**

a) Optimal – 2 page faults.

| 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **4** | 4 |
| | **2** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | **3** | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| | | | **4** | 4 | 4 | **5** | 5 | 5 | 5 | 5 | 5 |

b) LRU – 4 page faults.

| 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **5** |
| | **2** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | | **3** | 3 | 3 | 3 | **5** | 5 | 5 | 5 | **4** | 4 |
| | | | **4** | 4 | 4 | 4 | 4 | 4 | **3** | 3 | 3 |

c) FIFO – 6 page faults.

| 1 | 2 | 3 | 4 | 1 | 2 | 5 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | 1 | 1 | 1 | **5** | 5 | 5 | 5 | **4** | **4** |
| | **2** | 2 | 2 | 2 | 2 | 2 | **1** | 1 | 1 | 1 | **5** |
| | | **3** | 3 | 3 | 3 | 3 | 3 | **2** | 2 | 2 | 2 |
| | | | **4** | 4 | 4 | 4 | 4 | 4 | **3** | 3 | 3 |

d) CLOCK – 6 page faults ; second column is reference bit, star is the current position of the clock hand

| 1 | | 2 | | 3 | | 4 | | 1 | | 2 | | 5 | | 1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1* | 1 | 1* | 1 | 1* | 1 | 1* | 1 | 1* | 1 | 1* | 5 | 1* | 5 | 0 |
| | | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 0 | 1 | 1* |
| | | | | 3 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 | 0 | 3 | 0 |
| | | | | | | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 0 | 4 | 0 |

| 2 | | 3 | | 4 | | 5 | |
|---|---|---|---|---|---|---|---|
| 5 | 0 | 5 | 0 | 4 | 1* | 4 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 5 | 1* |
| 2 | 1* | 2 | 0 | 2 | 0 | 2 | 0 |
| 4 | 0 | 3 | 1* | 3 | 0 | 3 | 0 |

**Exercise 2:**

Assume that you have a page-reference string for a process with *m* frames (initially all empty). The page- reference string has length *p*; *n* distinct page numbers occur in it. Answer these questions for any page- replacement algorithms:

a. What is a lower bound on the number of page faults?

b. What is an upper bound on the number of page faults?

**Answer:**

a) There are *n* distinct pages occurring in the reference string, each of which must be loaded into memory at some point. Hence, the lower bound on the number of page faults is *n*, which is irrespective of the memory size, *m*.

b) The upper bound depends on the memory size, *m*. If $m \geq n$, then the upper bound will be *n* irrespective of *p*, since all the pages referenced in the string can be loaded into memory simultaneously. If $m < n$, the upper bound will be *p* because every page in the reference string will cause a fault.

**Exercise 3:**

Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of CPU and the paging disk. The results are one of the following alternatives. For each case, what is happening? Can the degree of multiprogramming be increased to increase the CPU utilization? Is the paging helping?

a) CPU utilization 13 percent; disk utilization 97 percent
b) CPU utilization 87 percent; disk utilization 3 percent
c) CPU utilization 13 percent; disk utilization 3 percent

**Answer:**

a) Thrashing is occurring, since the disk utilization is very high while the CPU is underutilized. Processes are spending most of their time paging. The degree of multiprogramming cannot be increased. On the contrary, one or more processes should be suspended to allow increase in the utilization of the CPU.
b) CPU utilization is sufficiently high to leave things alone. Any increase in the degree of multiprogramming could lead to thrashing.
c) The CPU is available for executing additional processes, so it would be wise to increase the degree of multiprogramming.