CS471: Operating System Concepts
Module 6: Homework #6
Points: 20

Question 1 [Points 9] Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

How many page faults would occur for the following replacement algorithms, assuming one, two, four, and seven frames?
Remember that all frames are initially empty, so your first unique pages will cost one fault each.

a. LRU replacement
b. FIFO replacement
c. Optimal replacement

Ans: One frame:
All 20 page references will result in a page fault---20 page faults---under all replacement policies

Two frames:
LRU: 1,2,3,4 cause page faults; 3,4 are in memory
2,1,5,6,2 cause page faults: 6,2 are in memory
1causes page fault: 1,2 are in main memory
2: NO page fault
3, 7, 6, 3, 2, 1 cause page faults: 2,1 are in main memory
2: No page fault
3, 6 cause page fault:

Total page faults: 18

FIFO: Same as LRU: 18 page faults

Optimal:
1,2 cause page fault: 1,2 in memory
3 causes page fault: 3,2 in memory
4 causes page fault: 4,2 in memory
2 NO Page fault: 4,2 in memory
1 causes page fault: 1,2 in memory
5 causes page fault: 5,2 in memory
6 causes page fault: 6,2 in memory
2 No page fault: 6,2 in memory
1 causes page fault: 1,2 in memory
2 NO page fault: 1,2 in memory
3 causes page fault: 3,2 in memory
7 causes page fault: 3,7 in memory
6 causes page fault: 3,6 in memory
3 causes NO page fault: 3,6 in memory
2 causes page fault: 3,2 in memory
1 causes page fault: 1,2 in memory
2 causes NO page fault: : 1,2 in memory
3 causes page fault: 3,2 in memory

6 causes page fault: 6,2 in memory

Total page faults in Optimal case=15

Four frames:
LRU:
1,2,3,4 cause page fault: 1,2,3,4 in memory
2,1 NO page fault: 1,2,3,4 in memory
5,6 cause page fault: 1,2,5,6 in memory
2,1,2 NO page fault: : 1,2,5,6 in memory
3,7 cause page fault: 1,2,3,7 in memory
6 causes page fault: 6,2,3,7 in memory
3,2 NO page fault: 6,2,3,7 in memory
1 causes page fault: 6,2,3,1 in memory
2,3,6 NO page fault: 6,2,3,1 in memory

Total page faults = 20-10=10

Reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

FIFO:
1,2,3,4 cause page fault: 1,2,3,4 in memory
2,1 NO page fault: 1,2,3,4 in memory
5,6 cause page fault: 5,6,3,4 in memory
2,1 cause page fault: : 5,6,2,1 in memory
2 NO page fault: 5,6,2,1 in memory
3,7 cause page fault: 3,7,2,1 in memory
6 causes page fault: 3,7,6,1 in memory
3 NO page fault: 3,7,6,1 in memory
2 causes page fault: 3,7,6,2 in memory
1 causes page fault: 1,7,6,2 in memory
2 NO page fault: 1,7,6,2 in memory
3 causes page fault: **1,3,6,2 in memory**
6 NO page fault: : **1,3,6,2 in memory**

Total page faults=20-6=14;

Optimal:
1,2,3,4 cause page fault: 1,2,3,4 in memory
2,1 NO page fault: 1,2,3,4 in memory
5 causes page fault: 1,2,3,5 in memory
6 causes page fault: 1,2,3,6 in memory
2,1,2,3 NO page fault: 1,2,3,6 in memory
7 causes page fault: 7,2,3,6 in memory
6,3,2, NO page fault: 7,2,3,6 in memory
1 causes page fault: 1,2,3,6 in memory
2,3,6 NO page fault: 1,2,3,6 in memory

Total page faults = 20-12=8

Seven frames:
In all cases, the following:
1,2,3,4 cause page fault: 1,2,3,4 in memory
2,1 NO page fault: 1,2,3,4 in memory
5 causes page fault: 1,2,3,4,5 in memory
6 causes page fault: 1,2,3,4,5,6 in memory
2,1,2,3 NO page fault: 1,2,3,4,5, 6 in memory
7 causes page fault: 1,2,3,4,5,6,7 in memory
6,3,2,1,2,3,6 NO page fault: 1,2,3,4,5,6,7 in memory

Total page faults: 20-13=7

Summary

	LRU	FIFO	**Optimal**
1	20	20	**20**
2	18	18	**15**
4	10	14	**8**
7	7	7	**7**

Question 2 [Points 5] 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 the CPU and the paging disk. Three alternative results are shown below. 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

Ans:

- **(a)** I/O is the bottleneck; more I/O-bound jobs in the job mix or possible thrashing. If degree of multiprogramming is low, then adding more CPU-bound jobs would increase the CPU utilization. If degree of multiprogramming is already high, it is due to thrashing and no more jobs should be added.
- **(b)** CPU is the bottleneck. I/O has poor utilization; multiprogramming should not be increased to further increase CPU utilization; we could add more I/O-bound jobs to the mix.
- **(c)** Both CPU and I/O utilizations are low; definitely adding more CPU-bound jobs will increase the CPU utilization.

Question 3 [Points 3] Consider a demand-paging system with a paging disk that has an average access and transfer time of 20 milliseconds. Addresses are translated through a page table in main memory, with an access time of 1 microsecond per memory access. Thus, each memory reference through the page table takes two accesses. To improve this time, we have added an associative memory that reduces access time to one memory reference if the page-table entry is in the associative memory. Assume that 80 percent of the accesses are in the associative memory and that, of those remaining, 10 percent (or 2 percent of the total) cause page faults. What is the effective memory access time?

Ans: Effective access time with Assoc memory = 0.8*1 + 0.18*2 + 0.02*(2+20*1000) =1.36+400.04=401.40 microseconds or 0.401 milliseconds

Question 4 [Points 3] Assume that we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty frame is available or if the replaced page is not modified and 20 milliseconds if the replaced page is modified. Memory-access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

Ans: Assume page-fault fraction to be p out of all memory accesses.

Effective access time = $(1-p)*100 + 0.3p*8*10^6 + 0.7p*20*10^6) = (1-p)*100 + p*(16.4*10^6)$
If effect. Access time $\leq 200 \Rightarrow (1-p)*100 + p*(16.4*10^6) \leq 200 \Rightarrow p*(16.4*10^6) \leq 100 \Rightarrow p \leq 100/(16.4*10^6) \Rightarrow p \leq 1/(16.4*10^4)$ or $p \leq 6.1*10^{-6}$

Maximum acceptable page fault rate is $6.1*10^{-6}$ or 0.00061%.