# **Exercises- Chapter 10 Virtual Memory**

## Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

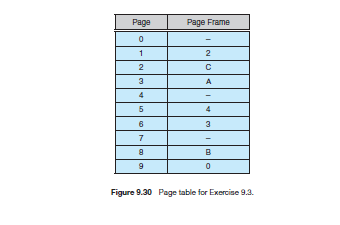
## Assume that you have a page-reference string for a process with *m* frames (initially all empty). The page-reference string has length *p*; *n* distinct page numbers occur in it. Answer these questions for any page-replacement algorithms:

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

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

## 3. Consider the page table shown in Figure 9.30 (next page) for a system with 12-bit virtual and physical addresses and with 256-byte pages. The list of free page frames is *D*, *E*, *F* (that is, *D* is at the head of the list, *E* is second, and *F* is last). Convert the following virtual addresses to their equivalent physical addresses in hexadecimal. All numbers are given in hexadecimal. (A dash for a page frame indicates that the page is not in memory.)

## .9EF • 111 • 700 • 0FF



4. Assume a program has just referenced an address in virtual memory.

Describe a scenario how each of the following can occur: (If a scenario

cannot occur, explain why.)

• TLB miss with no page fault

• TLB miss and page fault

• TLB hit and no page fault

# • TLB hit and page fault

# 5. Consider the following page-replacement algorithms. Rank these algorithms on a five-point scale from “bad” to “perfect” according to their page-fault rate. Separate those algorithms that suffer from Belady’s anomaly from those that do not.

# a. LRU replacement

# b. FIFO replacement

# c. Optimal replacement

# d. Second-chance replacement

# 6. Given 3 frames, find how many page faults occur with each of the algorithms below given the reference string

# 2 0 1 2 3 7 0 2 3 1 0 2 0 2 3 7 0 3 1 0 3 7

# a. LRU replacement

# b. FIFO replacement

# c. Optimal replacement

# d. Second-chance replacement

7. An operating system supports a paged virtual memory, using a central processor with a cycle time of 1 microsecond. It costs an additional 1 microsecond to access a page other than the current one. Pages have 1000 words, and the paging device is a drum that rotates at 3000 revolutions per minute and transfers 1 million words per second. The following statistical measurements were obtained from the system:

• 1 percent of all instructions executed accessed a page other than the

current page.

• Of the instructions that accessed another page, 80 percent accessed

a page already in memory.

• When a new page was required, the replaced page was modified 50

percent of the time.

Calculate the effective instruction time on this system, assuming that the system is running one process only and that the processor is idle during drum transfers.

8. Consider the two-dimensional array A:

int A[][] = new int[100][100];

where A[0][0] is at location 200 (i.e frame 1) in a paged memory system with pages of size 200. The matrix is stored as row-wise manner, i.e, first row, followed by second row etc..

A small process that manipulates the matrix resides in page 0 (locations 0 to 199). Thus, every instruction fetch will be from page 0.

Fora RAM with **three** frames, how many page faults are generated by the following array-initialization loops a) and b), using LRU replacement assuming that frame 0 contains the process and frames 1 and 2 contain the matrix?

a. for (int j = 0; j < 100; j++)

for (int i = 0; i < 100; i++)

A[i][j] = 0;

b. for (int i = 0; i < 100; i++)

for (int j = 0; j < 100; j++)

A[i][j] = 0;

9. 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