EE 463 **Operating Systems** **Date: 11/07/1444**
Winter 2023 **Homework #6** 02/02/2023
Dr. A. M. Al-Qasimi **Due: 09/02/2023**

Objectives:
The objective of this assignment is to exercise memory management techniques.

What to Do:
Read the Memory Management part (chapters 9 and 10) of the textbook *Operating System Concepts*, 10th edition, and then answer the following exercise questions.

1) Ex: 9.7
Assuming a 1 KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):
a. 3085
b. 42095
c. 215201
d. 650000
e. 2000001

2) Ex: 9.10
Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. How many entries are there in each of the following?
a. A conventional single-level page table?
b. An inverted page table?

3) Ex: 9.11
Explain the difference between internal and external fragmentation.

4) Ex: 9.13
Given six memory partitions of 100 MB, 170 MB, 40 MB, 205 MB, 300 MB, and 185 MB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 200 MB, 15 MB, 185 MB, 75 MB, 175 MB, and 80 MB (in order)? Indicate which—if any—requests cannot be satisfied. Comment on how efficiently each of the algorithms manages memory.

5) Ex: 9.26
What is the purpose of paging the page tables?

6) Ex: 10.9
Consider the following page reference string: 7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0 , 1.
Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms?
- LRU replacement
- FIFO replacement
- Optimal replacement

7) Ex: 10.15

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

8) Ex: 10.20

A certain computer provides its users with a virtual-memory space of 2^{32} bytes. The computer has 2^{22} bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4096 bytes. A user process generates the virtual address $(11123456)_h$. Explain how the system establishes the corresponding physical location. Distinguish between software and hardware operations.

9) Ex: 10.23

When a page fault occurs, the process requesting the page must block while waiting for the page to be brought from disk into physical memory. Assume that there exists a process with five user-level threads and that the mapping of user threads to kernel threads is many to one. If one user thread incurs a page fault while accessing its stack, will the other user threads belonging to the same process also be affected by the page fault—that is, will they also have to wait for the faulting page to be brought into memory? Explain.

10) Ex: 10.32

Suppose that a machine provides instructions that can access memory locations using the one-level indirect addressing scheme. What is the sequence of page faults incurred when all of the pages of a program are currently nonresident and the first instruction of the program is an indirect memory load operation? What happens when the operating system is using a per-process frame allocation technique and only two pages are allocated to this process?