

National University of Computer and Emerging Sciences, Lahore Campus



Course:	Operating Systems	Course Code:	CS 2006
Program:	BS(Computer Science)	Semester:	Spring 2022
Due Date	3-June-2022 at 11:30	Total Marks:	35
Type:	Assignment 3	Page(s):	2

Important Instructions:

1. Submit Soft copy of your solution using MS Word.
2. **Late submission of your solution is not allowed.**

Question 1: [5 marks]

Assume that the following three functions have already been implemented:

1. `int getPageSize() //returns page size in bytes.`
2. `int* getPageTable(int processId) //returns the page table of the process whose id is processId`
3. `int loadFrame(int processId, int pageNo) //loads the given pager number of the given process in memory and returns the frame number where the page has been loaded.`

When the process, whose id is processId, tries to access a virtual address (the address is stored in virtualAddr) belonging to a page that is not in memory, then a page fault exception will occur which will be handled by a page fault handler routine named as "handlePageFault". The purpose of this routine will be to load the page in memory. The signature of this routine is as follows:

```
void handlePageFault(int processId, int virtualAddr)
{
    //implement the routine
}
```

Your task is to implement this above routine with the help of routines already implemented.

Question 2: [10 marks]

Show execution of the LRU page replacement algorithm on the following page reference string: [10 marks]

2 3 1 4 1 5 3 2 1 4

Assume there are only three frames in the RAM. Show contents of memory frames after each page access from the reference string. (Please note that the number of boxes below maybe less or more depending upon the question. It, certainly, does not mean you have to utilize exactly the given number of boxes.)

Question 3: [10 marks]

Suppose that the page size is 128 bytes. Now consider process p1 whose page table is as follows:

0	5
1	10
2	9
3	6
4	Hard disk
5	1

Now assume that process p1 accesses the following virtual addresses:

a) 0 b) 139 c) 650 d) 257

1. Convert the above virtual addresses into physical addresses.
2. What is the range of virtual addresses of p1 that will cause page fault exception?

Question 4: [10 marks]

Consider the following code for a simple Stack:

```
class Stack {
private:
    int* a; // array for stack
    int max; // max size of array
    int top; // stack top
public:
    Stack(int m) {
        a = new int[m]; max = m; top = 0;
    }
    void push(int x) {
        while (top == max); // if stack is full then wait
        a[top] = x;
        ++top;
    }
    int pop() {
        while (top == 0); // if stack is empty then wait
        int tmp = top;
        --top;
        return a[tmp];
    }
};
```

Assuming the functions push and pop can execute concurrently, synchronize the code using semaphores. Also, replace the busy waiting with proper waiting.

GOOD LUCK FOLKS!

