**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCE**

**OPERATING SYSTEM QUIZ(B)**

**STUDENT ROLL NO: __________________ SECTION:______________**

**TIME ALLOWED: 20 MINUTES**

**<u>QUESTION # 1</u> (IDENTIFY WHETHER GIVEN STATEMENTS ARE TRUE OR FALSE. WRITE DOWN THE CORRECT STATEMENT AGAINST EACH FALSE STATEMENT)**

1) (T/F)Increasing main memory always decreases page fault rate.
   **(F) In FIFO replacement policy increasing main memory increases page fault rate.**

2) (T/F)Memory mapped technique is efficient to be used for randomly access file.
   **(T) Memory mapped technique is not good for a file that is sequential and it has to be used only one time.**

3) (T/F)Reducing the page size decreases the page table size.
   **(F) number of entries will be increased and it will increase page table size.**

4) (T/F)A smaller page size reduces paging I/O throughput.
   **(F) number of pages will be increased which might increase paging I/O throughput.**

5) (T/F)A blocking user-level thread blocks the process.
   **(T)**

6) (T/F)Interrupt disabling and enabling for enforcing mutual exclusion doesn't work on multiprocessor system.
   **(T)**

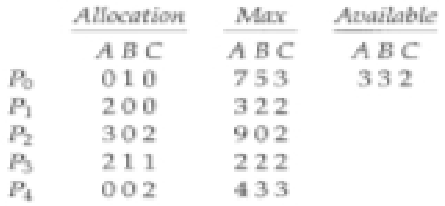
**<u>QUESTION # 2</u>**

Calculate number of page faults for the given string using LRU(3 page frames) ?
a b c d a b e a b c d e b a b

Page faults: _________________

**<u>QUESTION #3</u>**

Consider the snapshot of a safe system.



|  | *Allocation* | *Max* | *Available* |
|---|---|---|---|
|  | *A B C* | *A B C* | *A B C* |
| $P_0$ | 0 1 0 | 7 5 3 | 3 3 2 |
| $P_1$ | 2 0 0 | 3 2 2 |  |
| $P_2$ | 3 0 2 | 9 0 2 |  |
| $P_3$ | 2 1 1 | 2 2 2 |  |
| $P_4$ | 0 0 2 | 4 3 3 |  |

Can we immediately grant the request to process P0(1,1,3) .If yes, then provide a safe state(Assume processes are executed sequentially) .

**<u>QUESTION # 4</u>**

Does the given code fragment enforce mutual exclusion. If no, then identify the error in the given code fragment.

ANSWER: GIVEN CODE FRAGMENT DOESN'T GUARANTEE MUTUAL EXCLUSION. IF ONE READER ENTERS IN THE CRITICAL REGION IT ALLOWS SEVERAL WRITERS TO COME INSIDE THE CRITICAL REGION. ORDER OF SIGNAL (WRT) AND WAIT(WRT) SHOULD BE SWAPPED. AND ONLY FIRST AND LAST READER SHOULD BE ALLOWED TO DECREASE AND INCREASE THE VALUE OF WRT SEMAPHORE.

Semaphore mutex initialized to 1
Semaphore wrt initialized to 1
Integer readcount initialized to 0

```
WRITER
do {
 wait (wrt) ;
   //    writing is performed
 signal (wrt) ;
     } while (TRUE);
```

**READER**

```
    do {
                       wait (mutex) ;
                       readcount ++ ;
                         signal(wrt) ;
                    signal (mutex)
           // reading is performed
                    wait (mutex) ;
                         readcount  -- ;
                         wait(wrt) ;
                        signal (mutex) ;
      } while (TRUE);
```

**QUESTION # 5**

Consider a a page is stored in column major order. Which data structure provided below causes a low page fault rate and why?

**a)**

```
 i, j;
int[128][128] data;
for (j = 0; j < 128; j++)
for (i = 0; i < 128; i++)
data[i][j] = 0;
```

**b)**

```
int i, j;
int[128][128] data;
for (i = 0; i < 128; i++)
for (j = 0; j < 128; j++)
data[i][j] = 0;
```