THE UNIVERSITY OF DODOMA

COLLEGE OF INFORMATICS AND VIRTUAL EDUCATION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TEST 1 [25 MARKS]

SECOND SEMESTER 2020/2021

CP 226: OPERATING SYSTEMS

Date: th June **2021**

Time Allocated: 80 Minutes

Instructions:

1. *Answer all questions and be precisely*
2. *All University of Dodoma examination regulations apply.*

Name……………………………………………Reg#…………………………………

Question	**Marks**	**Score**
1	11	
2	14	
TOTOL	**25**	

Question One:

a. Briefly explain why implementing synchronization primitive by disabling interrupts is not appropriate in the computer systems with:-

   i. A single processor system if the synchronization primitive is to be used in user-level program. **[1 Mark]**

   If a user-level program is given the ability to disable interrupts, then it can disable the timer interrupt and prevent context switching from taking place, thereby allowing it to use the processor without letting other processes to execute.

   ii. Multiprocessor systems **[1 Mark]**

   Interrupts are not sufficient in multiprocessor systems since disabling interrupts only prevents other processes from executing on the processor in which interrupts were disabled; there are no limitations on what processes could be executing on other processors and therefore the process disabling interrupts cannot guarantee mutually exclusive access to program state.

b. Precisely show how a strict alternation mutual exclusion algorithm below either satisfies or fails to satisfy mutual exclusion and progress. **[3 Marks]**

```
Strict Alternation

      int turn = 0;

P0{
   while ( turn ! = 0);
   /*** Access shared data **/
   turn = 1;
}

P1 {
     while (turn ! =1);
     /*** Access shared data **/
     turn = 0;
    }
```

c. The reader-writer problem is one among classical synchronization problem in OS. Consider an airline reservation system in which many customers compete to write and read into and from the database. The system allow multiple customers (processes) to read simultaneously from the database but when one customer (process) is updating the database no other customers may have access to the database, not even the readers.. The aim of this problem is to be able to protect the shared data. Apply semaphore and mutex (combined) to implement a solution for the airline reservation database system. Make sure you handle busy waiting in your solution. **[6 Marks]**

Question Two

a. A resource allocation system that uses the Banker's algorithm for 3 resource types (A, B, C) and 5 users (P0, P1, P2, P3 P4) is currently in the following state as in Table 1. (Current Allocation: resources held by each user. Max Demand: max need of each user. Request: ongoing request of each user. Available: free resources.)

Table 1: Snapshot of the System

	Current Allocation			**Max. Demand**			**Request For Each Process**			**Available**		
Process	**A**	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
P_0	0	1	0	7	5	3	3	2	2	3	3	2
P_1	2	0	0	3	2	2	0	2	1			
P_2	3	0	2	9	0	2	6	0	0			
P_3	2	1	1	2	2	2	0	1	0			
P_4	0	0	2	4	3	3	2	3	0			

i. Calculate the Need Matrix **[2 Marks]**

	A	**B**	**C**
P_0	**7**	**4**	**5**
P_1	**1**	**2**	**2**
P_2	**6**	**0**	**0**
P_3	**0**	**1**	**1**
P_4	**4**	**3**	**1**

ii. Is the state safe. If yes, give a sequence of process ids that leads to all processes execution to be completed. If no, give a sequence of activities that result in a deadlocked state. Note that you must show how the available (working array) changes as each process terminate.

[4 Marks]

Yes, the state is safe. There are many safe sequences. Here is one: < P1, P3, P2, P4, P0 > is a safe sequence. There is no safe sequence starting with P0. The state is not deadlocked (obviously, since it is safe). **(2 Marks)**

How the working array Changes

	A	B	C
	3	**3**	**2**
P_1	**5**	**3**	**2**
P3	**7**	**4**	**3**
P_2	**10**	**4**	**5**
P_4	**10**	**4**	**7**
P0	**10**	**5**	**7**

(2 Marks)

iii. Is there an ongoing request other than **P_2's** that cannot be granted immediately? Justify your answer. **[2 Marks]**

YES

P4's request cannot be granted in the above state. Doing so

would lead to Avail = [1 0 2] and Need = [2 0 1], which is not a safe state.

b. Five batch jobs A through E arrive at a computer center in the order A to E, at almost the same time. They have estimated running times of 6, 4, 1, 3, and 7 minutes. Their priorities are 3, 5, 2, 1, and 4, respectively, with 5 being the highest priority. For each of the following scheduling algorithms, (Round Robin (Quantum = 1) and Priority scheduling):- **[6 Marks]**

i. Show the scheduling order for these processes.

With RR with Quantum 1

Gnatt chart is:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A	B	C	D	E	A	B	D	E	A	B	D	E	A	B	E	A	E	A	E	E	

[1 Marks]

With Priority scheduling

The processes are scheduled in the order: **B E A C D**

[1 Marks]

ii. Determine the mean process waiting time and turnaround time. Ignore process switching overhead.

Mean waiting Time and Turnaround Time with RR

T_{wait} = ((0+4+3+3+2+1) + (1+4+3+3) + (2) + (3+3+3) +

(4+3+3+2+1+1+0) / 5

= (13 + 11 + 2 + 9 + 14) /5

= 49 / 5

= **9.8** (minutes) **[1 Marks]**

T(arround_Time) **average = 21.2** **[1 Marks]**

Mean waiting Time and Turnaround Time Priority

T_{wait} = (11 + 0 + 17 + 18 + 4) /5

= 50 / 5

= **10** (minutes) **[1 Marks]**

Turnaround Time **average** = **20** **[1 Marks]**