Operating System Solution Spring Semester -2020

10th JUNE 2020 & 8 AM to 10 PM For CSE & CSCE Branch Question of SLOT – 1 (Solution to be done by)	
Section A Question No. 1 (A & B)	Prof. Bindu Agarwalla
Section A Question No. 2 (A & B)	Prof. Tanmoy Maitra
Section B Question No. 3 (A & B)	Prof. (Dr.) Alok Kumar Jagadev Already submitted with the question (No need to submit again)
Section B Question No. 4 (A & B)	Prof. Himansu Das
Section B Question No. 5 (A & B)	Prof. (Dr.) Biswajit Sahoo

SECTION - A

1. (a) Justify whether "Creation of multiple threads is preferred over multiple processes or not". [2.5 M arks]

SLOT- 1 (CS & CSSE)

Ans 1.a) Creation of multiple threads is preferred over multiple processes, as —

1) Creation of multiple processes is more resource intensive as compared to threads.

2) Context switching is faster with threads than processes.

3) Communication among multiple threads is easier, as the threads shares common addr space, but with processes, specific comm technique need to be followed

4) Threads provide enhanced throughput of the sys.

1. (B)

Consider process arrival as given below where n = right most significant digit of your Roll No.(ex:- for Roll No. 180854, n=**4**): . [10 Marks]

Process	CPU Burst Time(ms)	Arrival Time
A	4	0
B	3	6
C	6	n
D	1+(n%9)	3
E	1	4

Calculate the following for *round robin* (time quantum = 2 ms) CPU scheduling algorithm:

i. Average waiting time

ii. Turnaround time for each process

iii. Order of completion

iv. Minimum time quantum so that *round robin* will act as *first come first serve* scheduling

ANS.

SLOT-1

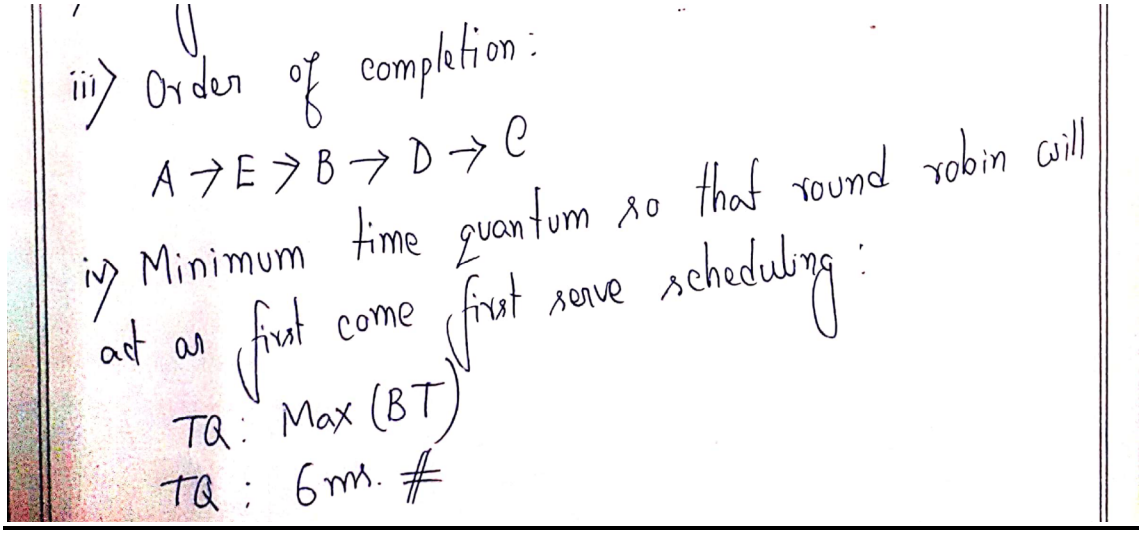
1.b) For Roll No 1805854 (ii)

Process	CPU BT (ms)	AT	CT	TAT	WT
A	4	0	4	4	0
B	3	6	16	10	7
C	6	4	19	15	9
D	1+(n%9)	3	17	14	9
E	1	4	9	5	4

TQ = 2ms.

A	A	D	C	E	B	D	C	B	D	C
0	2	4	6	8	9	11	13	15	16	17	19

i) Avg. WT = 5.8 ms.



Calculation of CPU burst time of of process D Considering the process arrival as "n" given in the question where n = right most significant digit of your Roll No.(ex:- for Roll No. 180854, n=4)

Value of "n"	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
CPU burst time of D as "1+(n%9)"	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2. (A) Write down WAIT( ) and SIGNAL( ) operations on counting semaphore. [2.5]

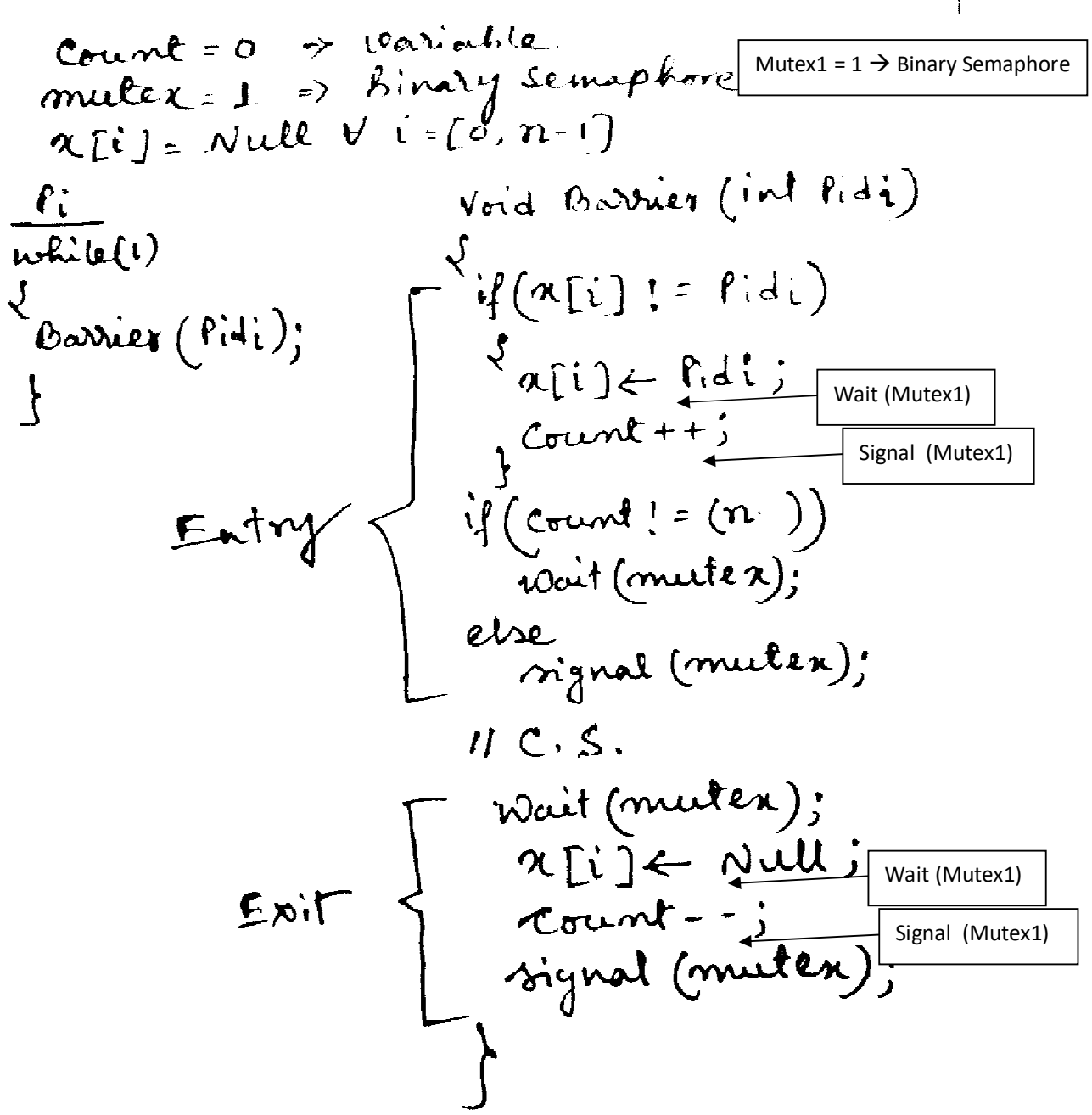
ANS.

Explain WAIT( ) and SIGNAL( ) operations → 1 Mark

If code given with respect to counting semaphore → 1.5 Marks

2. (B) Assume a set *X of n* processes, where $n < 20$ with process IDs [0, *n*-1]. A function *void Barrier(int pid)* is defined over the set *X* such that if anyone process P_0 in *X* with a process ID pid_0 calls *Barrier(int* pid_0*)*, it is blocked until all processes in *X* have called *Barrier( )* with their process IDs. Write the pseudo-code for the *Barrier( )* function using only semaphores. List the semaphores that you have used with their initial values without concerning who creates and initializes. Show the code for *Barrier( )* that uses the semaphores declared by making WAIT( ) and SIGNAL( ) calls on them. [10 Marks]

ANS.



SECTION - B

3. (A) In a system n processes, P_1, ... , P_n share m resources of same type, R. Each resource can be acquired and released one at a time. If the maximum requirement of resources for each process P_i is r_i, where $r_i > 0$, then find the sufficient condition in extreme situation to ensure deadlock does not occur in the system. [4.5 Marks]

ANS.

In extreme situation, each process P_i holding (r_i-1) number of resources and request for one more resource. So the following condition must be true to make sure that deadlock never occurs.

So,

$$\sum_{i=1}^{n}(r_i - 1) < m$$

$$\left(\sum_{i=1}^{n} r_i - \sum_{i=1}^{n} 1\right) < m$$

$$\sum_{i=1}^{n} r_i - n < m$$

$$\sum_{i=1}^{n} r_i < m + n$$

3. (B) An operating system uses the Banker's algorithm for deadlock avoidance when managing the allocation of three resource types R_0, R_1, and R_2 to three processes P_0, P_1, and P_2. The snapshot of current system state is shown in the following table. The Allocation matrix shows the number of resources of each type already allocated to each process and the Max matrix shows the maximum number of resources of each type required by each process during its execution. [8 Marks]

Process	Max				Allocation		
	R_0	R_1	R_2		R_0	R_1	R_2
P_0	7	3	3		1	0	1
P_1	5	2	1		2	2	1
P_2	6	3	3		2	1	1

There are 3 instances of resource type R_0, 2 instances of resource type R_1 and 2 instances of resource type R_3 still available in the system. Check the system is currently in a safe state. Find out which process will complete its execution at end? Consider the following independent requests for additional resources in the current state:

REQ1: P0 requests 2 instances of resource type R3 only.

REQ2: P1 requests 1 instances of resource type R0 only.

Explain, which of the above request can be permitted?

ANS.

	Max				Allocation				Need				Available		
	R_0	R_1	R_2		R_0	R_1	R_2		R_0	R_1	R_2		R_0	R_1	R_2
P_0	7	3	3		1	0	1		6	3	2		3	2	2
P_1	5	2	1		2	2	1		3	0	0				
P_2	6	3	3		2	1	1		4	2	2				

Safe state: $<P_1, P_2, P_0>$, P_0 will be executed at end.

Now, if the request REQ1 is permitted, the state would become:

	Max				Allocation				Need				Available		
	R_0	R_1	R_2		R_0	R_1	R_2		R_0	R_1	R_2		R_0	R_1	R_2
P_0	7	3	3		1	0	3		6	3	0		3	2	0
P_1	5	2	1		2	2	1		3	0	0				
P_2	6	3	3		2	1	1		4	2	2				

REQ1 will not be permitted.

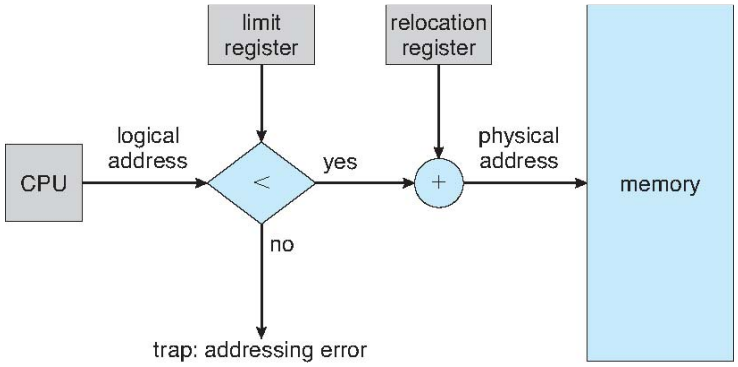
Now, if the request REQ2 is permitted, the state would become:

	Max				Allocation				Need				Available		
	R_0	R_1	R_2		R_0	R_1	R_2		R_0	R_1	R_2		R_0	R_1	R_2
P_0	7	3	3		1	0	1		6	3	0		2	2	2
P_1	5	2	1		3	2	1		3	0	0				
P_2	6	3	3		2	1	1		4	2	2				

REQ 2 will be permitted.

4. (A) How can not a process be allowed to access the memory locations that it does not have its own? [4.5 mark]

ANS:- process can't allowed to access the memory locations that it doesn't have its own by using relocation register and limit register. The relocation register contains the value of the smallest physical address and limit register contains the range of logical addresses Each logical address must fall within the range specified by the limit register. The MMU maps the logical address dynamically by adding the value in the relocation register. This mapped address is send to memory for fetching the information.



4. (B) Compare the address translation mechanism of paging and segmentation for convection of logical addresses to physical addresses. Consider a system with 32 bits logical address and 4KB of the page size. The system has also 512MB of physical memory. Find the following

i. Number of entries in a single-level page table

ii. Size of the single-level page table

iii. Maximum size of the program for two level paging?

[8 marks]

B) **ANS:-**

Paging	Segmentation
1. Logical memory is partitioned into pages.	1. Logical memory is partitioned into segments.
2. OS maintains a page table for mapping between frames and pages.	2. OS maintains a segment table for mapping between frames and pages.
3. It doesn't support user view of memory	3. It supports user view of memory
4. Page number and offset are used to compute absolute address.	4. Segment number and offset are used to compute absolute address.

Length of the Logical address=32 bits

Page size = 4 KB =2^{12}

Offset=12 bits

Page no. =20 bits

Physical Memory=512 MB=2^{29}

No. Of frames=$\frac{2^{29}}{2^{12}}=2^{17}$

(i) Number of entries in a single-level page table is 2^{20}

(ii) Size of the single level page table is $2^{20} \times 17$ bits

(iii) Maximum size of the program for two level paging is 2^{32} =4 GB

5. (A) A manufacturer wishes to design a hard disk with a capacity of M GB. If the technology used to manufacture the disks allows 1024-byte sectors, 2048 sectors/track, and 4096 tracks/platter, how many platters are required? (Assume M=10*(1+(right most digit of your Roll No % 9)); Ex:- for Roll No. 183452; M=10*(1+(**2** % 9)) i. e 30) [4.5 Marks]

ANS:

No of bytes in a platter = 4096 * 2048 *1024 B

$=2^{12} * 2^{11} * 2^{10}$ B

$=2^{3} * 2^{30}$ B

=8GB

For Roll No. 183452; M=10*(1+(**2** % 9))

M= 30

Now, No. of platter required for 30GB disk = 30GB / 8GB

=3.75

So, Number of platters = 4 (Ans)

Generalized Answer

RMD of Roll No	0	1	3	4	5	6	7	8	9

Answer	2	3	5	7	8	9	10	12	2

5. (B)

(b) Explain the purpose and importance of system calls related to device management. Discuss the structure of directory and its implementation in detail. [8.0]

ANS:

Theory answer available in Text Book.

4 marks for first part

4 marks for second part