Summary in Graph

Exam Summary (GO Classes Test Series 2024 | Operating Systems | Test 2)

Qs. Attempted:	**11** 4 + 7	**Correct Marks:**	**11** 1 + 10
Correct Attempts:	**6** 1 + 5	**Penalty Marks:**	**1** 0.33 + 0.67
Incorrect Attempts:	**5** 3 + 2	**Resultant Marks:**	**10** 0.66 + 9.33

Total Questions:	**15** 5 + 10
Total Marks:	**25** 5 + 20
Exam Duration:	45 Minutes
Time Taken:	45 Minutes

EXAM RESPONSE | EXAM STATS | FEEDBACK

Technical

Q #1 | Multiple Select Type | Award: 1 | Penalty: 0 | Operating System

In order to create a good solution for the mutual exclusion problem(critical section problem) for concurrent processes, which of the following conditions must hold?

A. No process should have to wait forever to enter its critical region.
B. No process running outside of its critical region may block other processes from entering their critical region.
C. There should be no assumptions about the speed or number of CPUs.
D. Every process should be able to enter inside the critical section as soon as it shows interest.

Your Answer: | Correct Answer: A;B;C | Not Attempted | Discuss

Q #2 | Multiple Select Type | Award: 1 | Penalty: 0 | Operating System

Consider a computer system in which processes can request and release one or more resources. Once a process has been granted a resource, the process has exclusive use of that resource until it is released. If a process requests a resource that is already in use, the process enters a queue for that resource, waiting until the resource is available. Which of the following will deal with the problem of deadlock such that deadlock is prevented or handled?

A. Giving priorities to processes and ordering the wait queues by priority
B. Having a process request all its required resources when it first begins, and restarting if it cannot obtain them all
C. Numbering the resources and requiring that processes request resources in order of increasing number
D. Having processes time out and restart after a random interval of waiting

Your Answer: A;B;D | Correct Answer: B;C;D | Incorrect | Discuss

Q #3 | Multiple Select Type | Award: 1 | Penalty: 0 | Operating System

Which of the following statements about deadlocking is/are true?

A. If a system is not deadlocked at time T, it can always avoid being deadlocked at time $T + 1$.
B. If a system is in a safe state at time T, it can always avoid being deadlocked at time $T + 1$.
C. Cycle in the resource allocation graph always lead to deadlock.
D. If the Banker's algorithm will not approve a resource request, and the resource request is still processed, then system necessarily will enter deadlock.

Your Answer: B;D | Correct Answer: B | Incorrect | Discuss

Q #4 | Multiple Choice Type | Award: 1 | Penalty: 0.33 | Operating System

Suppose there are 'P' processes, all the processes share a total of 'R' identical resources, which can be reserved and released one at a time. Each process requires a maximum of 'M' resource units, where $M > 0$. Which one of the following is a necessary condition for ensuring deadlock will not occur?

A. $P(M - 1) > R - 1$
B. $P(M - 1) \leq R - 1$
C. $P(M - 1) < R - 1$
D. $P(M - 1) \geq R - 1$

Your Answer: C | Correct Answer: B | Incorrect | Discuss

Q #5 | Multiple Choice Type | Award: 1 | Penalty: 0.33 | Operating System

A cycle in a resource-allocation graph is ______

A. a necessary and sufficient condition for deadlock in the case that each resource has more than one instance
B. a necessary and sufficient condition for a deadlock in the case that each resource has exactly one instance
C. a sufficient condition for a deadlock in the case that each resource has more than once instance
D. is neither necessary nor sufficient for indicating deadlock in the case that each resource has exactly one instance

Your Answer: B | Correct Answer: B | Correct | Discuss

Q #6 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | Operating System

Consider the following 3 processes with 3 binary semaphores with initial values $s0 = 0, s1 = 0, s2 = 1$

$$\begin{array}{lll} \textbf{P:} & \textbf{Q:} & \textbf{R:} \\ \text{While}(1) & \text{While}(1) & \text{While}(1) \\ \{ & \{ & \{ \\ \quad \text{P(s0);} & \quad \text{P(s1);} & \quad \text{P(s2);} \\ \quad \text{Print (0);} & \quad \text{Print(1);} & \quad \text{Print(2);} \\ \quad \text{V(s1);} & \quad \text{V(s1);} & \quad \text{V(s0);} \\ \} & \} & \} \end{array}$$

What will be the correct pattern generated by these 3 processes?

A. $(201)^*$
B. $(012)^*$
C. $\left(201^+\right)^*$
D. 201^+

Your Answer: A | Correct Answer: D | Incorrect | Discuss

Q #7 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | Operating System

D number of friends go to a Chinese restaurant at a busy time of the day. The waiter apologetically explains that the restaurant can provide only a few chopsticks to be shared among the D people. Furthermore, each diner may require a different number of chopsticks to eat. For example, it is possible that one of the diners is an octopus, who for some reason refuses to begin eating before acquiring eight chopsticks. The second parameter of this scenario is C, the number of chopsticks that would simultaneously satisfy the needs of all diners at the table. For example, two octopuses would result in $\mathrm{C} = 16$.
All the chopsticks provided by the waiter are placed in an empty glass at the center of the table and each diner obeys the following protocol:
Acquire chopsticks one after one and start eating only after having the required number of chopsticks. If no more chopsticks are available then Hold the current chopsticks and wait for more chopsticks. Once all chopsticks are acquired, then eat and release all chopsticks. Also while one person is using a chopstick then unless he releases it after completion of eating, it can not be shared with any other person(i.e. Chopsticks are used in mutual exclusion manner).

What is the smallest number of chopsticks (in terms of D and C) needed to ensure that deadlock cannot occur?

A. C
B. D
C. $\mathrm{C} - \mathrm{D}$
D. $\mathrm{C} - \mathrm{D} + 1$

Your Answer: | Correct Answer: D | Not Attempted | Discuss

Q #8 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | Operating System

Consider the following concurrent tasks, in which each assignment statement executes atomically. Initially, the shared variables x and y are set to 0.

$\underline{\text{Task 1}}$

$$x = 1$$
$$a = y$$

$\underline{\text{Task 2}}$

$$y = 1$$
$$b = x$$

At the end of the concurrent tasks, the values of a and b are examined. Which of the following must be true?

I. $(a == 0) \rightarrow (b == 1)$
II. $(b == 0) \rightarrow (a == 1)$
III. $(a == 1) \rightarrow (b == 1)$

A. I only
B. II only
C. III only
D. I and II only

Your Answer: D | Correct Answer: D | Correct | Discuss

Q #9 | Multiple Select Type | Award: 2 | Penalty: 0 | Operating System

Consider the following snapshot of a system with five processes $(\mathrm{P1}, \mathrm{P2}, \mathrm{P3}, \mathrm{P4}, \mathrm{P5})$ and four resources $0(\mathrm{R1}, \mathrm{R2}, \mathrm{R3}, \mathrm{R4})$. There are no current outstanding queued unsatisfied requests.

Currently Available Resources

R1	R2	R3	R4
2	1	2	0

	Current Allocation				Max Need				Still Needs			
Process	**R1**	**R2**	**R3**	**R4**	**R1**	**R2**	**R3**	**R4**	**R1**	**R2**	**R3**	**R4**
P1	0	0	1	2	0	0	3	2	0	0	2	0
P2	2	0	0	0	2	7	5	0	0	7	5	0
P3	0	0	3	4	6	6	5	6	6	6	2	2
P4	2	3	5	4	4	3	5	6	2	0	0	2
P5	0	3	3	2	0	6	5	2	0	3	2	0

Consider the following statements.

From the perspective of deadlock avoidance, which one of the following is false?

A. If a request from process $\mathrm{P1}$ arrives for $(0,0,2,0)$, the request can be immediately granted because the system will remain in safe state after granting this request.
B. If a request from process $\mathrm{P3}$ arrives for $(0,0,2,0)$, the request can be immediately granted because the system will remain in safe state after granting this request.
C. If a request from process $\mathrm{P5}$ arrives for $(0,0,2,0)$, the request can be immediately granted because the system will remain in safe state after granting this request.
D. If a request from process $\mathrm{P2}$ arrives for $(0,0,2,0)$, the request can be immediately granted because the system will remain in safe state after granting this request.

Your Answer: | Correct Answer: B;C;D | Not Attempted | Discuss

Q #10 | Multiple Select Type | Award: 2 | Penalty: 0 | Operating System

Consider the following algorithm for a solution to the critical section problem for two processes. The solution for process $\mathrm{P_i}$ $(\mathrm{i} = 0 \text{ or } 1)$ with $\mathrm{P_j}$ $(\mathrm{j} = 1 \text{ or } 0)$ is shown below:

```
    boolean flag[2];
    flag[0] = 0, flag[1]=0;
    int turn = 0;

    Code for P_i:
    repeat forever
    flag[i] = 1;
    while (flag[j])
    {
        if (turn == j)
        {
            flag[i] = 0;
            while(turn == j);
            flag[i] = 1;
        }
    }
    /* enter C.S. */
    /* exit C.S. */
    turn = j;
    flag[i] = 0;
```

Which of the following is true?

A. Mutual exclusion is satisfied.
B. Progress is satisfied.
C. Bounded waiting is satisfied.
D. Deadlock may occur.

Your Answer: A;D | Correct Answer: A;B;C | Incorrect | Discuss

Q #11 | Multiple Select Type | Award: 2 | Penalty: 0 | Operating System

Consider a queue between the two processes indicated below. N is the length of the queue; e, f, and b are semaphores.

```
    init: e:=N ; f:=0 ; b:=1;
```

Process 1 :	Process 2 :
loop	loop
P(e)	P(f)
P(b)	P(b)
enqueve	dequeue
V(b)	V(b)
V(f)	V(e)
end loop	end loop

Which of the following statements is (are) true?

A. The purpose of semaphore f is to ensure that dequeue is not executed on an empty queue.
B. The purpose of semaphore e is to ensure that deadlock does not occur.
C. The purpose of semaphore b is to provide mutual exclusion for queue operations.
D. Deadlock is possible.

Your Answer: A;C | Correct Answer: A;C | Correct | Discuss

Q #12 | Multiple Select Type | Award: 2 | Penalty: 0 | Operating System

Suppose two threads execute the following C code concurrently, accessing shared variables a, b, and c :

Initialization

```
int a = 4;
int b = 0;
int c = 0;
```

Thread 1

```
if (a < 0) {
    c = b - a;
} else {
    c = b + a;
}
```

Thread 2

```
b = 10;
a = -3;
```

You can assume that reads and writes of the variables are atomic, and that the order of statements within each thread is preserved in the code generated by the C compiler.

Statement $\mathrm{X} = \mathrm{Y} * \mathrm{Z}$ reads Y, then reads Z from memory, performs $\mathrm{Y} * \mathrm{Z}$, stores/writes it to memory X.

Which of the following values are possible for c after both threads complete?

A. 4
B. 7
C. 14
D. -3

Your Answer: A;B;C;D | Correct Answer: A;B;C;D | Correct | Discuss

Q #13 | Multiple Select Type | Award: 2 | Penalty: 0 | Operating System

Consider the following pseudo-code for a process Pi, where "**shared boolean flag[2]**" is a variable declared in shared memory, initialized as:

```
flag[0] = flag[1] = FALSE;
P(int i) { // i=0 for P0, i=1 for P1
    while (TRUE) {
        flag[i]= TRUE;
    while (flag[(i+1) of 2]== TRUE);
        < Critical section >
    flag [i] = FALSE;
        < Remainder Section >
    }
}
```

this code tries to solve the critical section problem for two processes P0 and P1.

Which of the following is/are true?

A. Deadlock is possible.
B. Mutual exclusion is satisfied.
C. Bounded waiting is satisfied.
D. Progress is satisfied.

Your Answer: | Correct Answer: A;B;C | Not Attempted | Discuss

Q #14 | Multiple Select Type | Award: 2 | Penalty: 0 | Operating System

Consider the following snapshot of a system with five processes $(P10, P11, P12, P13, P14)$ and four resources $(R1, R2, R3, R4)$. $R1, R2, R3$, and $R4$ have a total of $6, 7, 14$, and 12 resources, respectively. There are no current outstanding queued unsatisfied requests.

Table 1: Allocation

PID	R1	R2	R3	R4
10	0	0	1	2
11	2	0	0	0
12	0	0	3	4
13	2	3	5	4
14	0	3	3	2

Table 2: Max Need

PID	R1	R2	R3	R4
10	0	0	3	2
11	2	7	5	0
12	6	6	5	6
13	4	3	5	6
14	0	6	5	2

From the perspective of deadlock avoidance, which one of the following is true?

A. The system is in safe state
B. The system is not in safe state, but would be if one more instance of $R1$ were available
C. If a request from process $P14$ arrives for $(0, 2, 2, 0)$, the request can be immediately granted.
D. If a request from process $P14$ arrives for $(0, 2, 2, 0)$, the request cannot be immediately granted.

Your Answer: A;D **Correct Answer: A;D** Correct Discuss

Q #15 Numerical Type Award: 2 Penalty: 0 Operating System

Four processes A, B, C, D are running concurrently using a shared variable counter which is initialized to zero. Processes A and B read count from memory, increment it by one and store back into memory and terminates. Processes C and D read count from memory and decrement it by two and store it back into memory and terminates. To synchronize the execution of the 4 processes, a semaphore S is used which is initialized to 2. See the table below.

Process A	Process B	Process C	Process D
WAIT (S) Read (count) count=count+1 Write (count) SIGNAL (S)	WAIT (S) Read(count) count=count+1 Write (count) SIGNAL (S)	WAIT(S) Read(count) count=count-2 Write (count) SIGNAL (S)	WAIT(S) Read(count) count=count-2 Write (count) SIGNAL (S)

If the maximum and minimum possible values of “count” after all processes complete execution are X, Y respectively then $X - Y$ is _______

Your Answer: 6 **Correct Answer: 6** Correct Discuss