2022_CS341-Operating System Quiz-2

Total points 34/90

Email *

tarusimittal@gmail.com

✗ Synchronization with....., you don't need to grab a mutex * 0/2

- (●) Semaphore ✗
- ( ) Monitor
- ( ) Condition Variable
- ( ) Mutual Exclusion

✓ * 1/1

```
A semaphore whose definition includes the fairest policy First-in-First-Out (FIFO) is called a ……..
A) binary semaphore
B) strong semaphore
C) weak semaphore
D) multi semaphore
```

- ( ) A
- (●) B ✓
- ( ) C
- ( ) D

✓ Which of the following scheduling algorithms could result in starvation? * 2/2

- ○ First-come, first-served
- ◉ Shortest job first ✓
- ○ Round robin
- ○ Priority

✗ For implementing locks, ..., works on both uniprocessors and multiprocessors. * 0/3

- ○ Test&set
- ◉ Spin Lock ✗
- ○ Either Test&Set Spin Lock
- ○ None of the above

✗ ..... is a strategy by which a user (or an application) exploits the characteristics of the CPU scheduling policy to get as much of the CPU time as possible. * 0/1

- ◉ synchornization ✗
- ○ Round robin
- ○ time-sharing
- ○ countermeasure

✗ A program consists of a single loop that executes 50 times. The loop contains a computation that consumes 50 ms of CPU time, followed by an I/O operation that lasts for 200 ms.The program is executed in a multiprogramming OS with negligible overhead. Assuming the program has the highest priority in the system, compute its elapsed time? Given answer as x.y ms * ···/2

1000 ✗

✗ * 0/2

---------- Enforces mutual exclusion, while providing effective means of inter-process communication.
A) Semaphores
B) Messages
C) Monitors
D) Addressing

- (●) A ✗
- ( ) B
- ( ) C
- ( ) D

✗ In the following code, three processes produce output using the routine 'putc' and synchronise using semaphores "L" and "R". Is CABABDDCABCABD a possible output sequence when this set of processes runs? * 0/2

```
semaphore L = 3, R = 0;  /* initialization */

/* Process 1 */          /* process 2 */          /* process 3 */
L1:                      L2:                      L3:
    P(L);                    P(R);                    P(R);
    putc('C');               putc('A');
    V(R);                    putc('B');                   putc('D');
                             V(R);
    goto L1;                 goto L2;                 goto L3;
```

- (●) Yes ✗
- ( ) No

✓ * 1/1

Banker's algorithm is
A) Deadlock prevention
B) Deadlock avoidance
C) Deadlock ignorance
D) Deadlock detection

- ( ) A
- (●) B ✓
- ( ) C
- ( ) D

✓ .....is a queue of threads waiting for something inside a critical section * 2/2

- ( ) Synchronisation Queue
- ( ) MLFQ
- (•) Condition Variable ✓
- ( ) Mutual Exclusion

✗ Consider the following mix of jobs, job lengths, and arrival times, assume a time slice of 10 and compute average response time (assuming RR scheduling algo) is * ···/2

Job	length	arrival time
0	75	0
1	40	10
2	25	10
3	20	80
4	45	85

76 ✗

Find the best match *

	approximate SRTF	busy waiting	Java	synchronous events in the machine	tasks are placed into the lowest-priority queue	Threads	Score	
Monitor	☑	☐	☐	☐	☐	☐	0/1	✗
Traps	☐	☑	☐	☐	☐	☐	0/1	✗
spinlocks	☐	☐	☑	☐	☐	☐	0/1	✗
MLFQ	☐	☑	☐	☐	☐	☐	0/1	✗

✓ Linux, uses spinlocks as a synchronization mechanism only on multiprocessor systems * 1/1

- (●) True ✓
- ( ) False

✓ *

2/2

At a particular time of computation, the value of a counting semaphore is 7. Then 20 P operations and 'x' V operations were completed on this semaphore. If the final value of the semaphore is 5, x will be

A. 22
B. 18
C. 15
D. 13

- ○ A
- ◉ B ✓
- ○ C
- ○ D

✗ Consider the following mix of jobs, job lengths, and arrival times, assume a time slice of 10 and compute average response time (assuming RR scheduling algo) is * ···/2

Job	length	arrival time
0	75	0
1	40	10
2	25	10
3	20	80
4	45	85

76 ✗

✓ 5 jobs, of length 50, 40, 30, 20, and 10 seconds each, time slice 1 second, context switch time of 0 seconds. Assuming RR, average completion time is * 2/2

Job	Length
1	50
2	40
3	30
4	20
5	10

110 ✓

✓ *

1/1

In resource allocation denial, a …….. is one in which there is at least one sequence that does not result in a deadlock.
A) Safe state
B) Unsafe state
C) Safe allocation
D) Unsafe allocation

- ○ D
- ○ C
- ○ B
- ◉ A ✓

✓ 5 jobs, of length 50, 40, 30, 20, and 10 seconds each, time slice 1 second, context switch time of 0 seconds. Assuming RR completion time of Job 5 is * 2/2

Job	Length
1	50
2	40
3	30
4	20
5	10

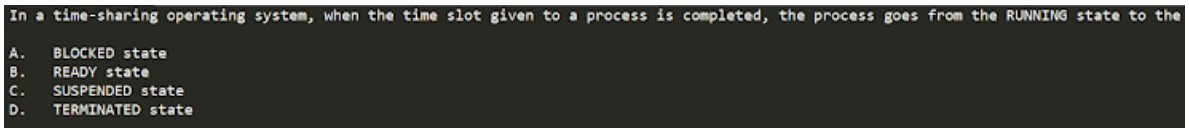
50 ✓

Find the best match *

	stop one of four necessary conditions for deadlock	Assesses, for each allocation, whether it has the potential to lead to deadlock	Attempts to assess whether waiting graph can ever make progress	Ignore the problem and pretend that deadlocks never occur in the system	Allow system not to enter deadlock	Score	
Techniques for addressing Deadlock	☑	☐	☐	☐	☐	0/1	✗
Deadlock prevention	☑	☐	☐	☐	☐	1/1	✓
Deadlock avoidance	☑	☐	☐	☐	☐	0/1	✗
Deadlock detection (next time) and recover	☑	☐	☐	☐	☐	0/1	✗

✓ *

2/2



- ○ A
- ◉ B ✓
- ○ C
- ○ D

✓ Starvation implies deadlock *

1/1

- ○ True
- ◉ False ✓

✓ *

1/1

..... when a process leaves a critical section and more than one process is waiting, the selection of a waiting process is arbitrary.
A) Busy waiting is employed
B) Starvation is possible
C) Deadlock is possible
D) All of the above

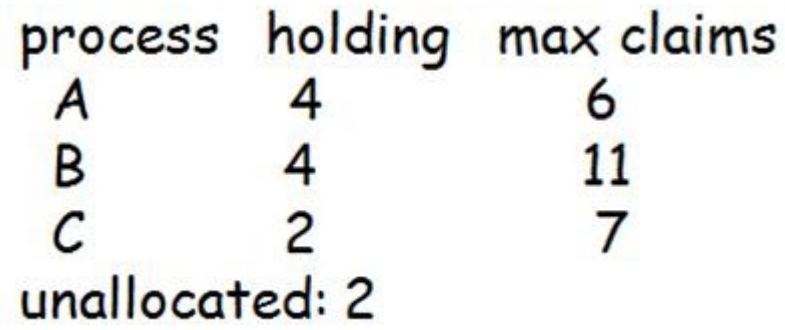
- ○ A
- ◉ B ✓
- ○ C
- ○ D

✗ From the following mix of jobs, job lengths, and arrival times, assume a time slice of 10 and compute the completion time of Job 0 (assuming FIFO scheduling algo) is * 0/2

Job	length	arrival time
0	75	0
1	40	10
2	25	10
3	20	80
4	45	85

205 ✗

✗ Assuming one resource class. If safe sequence of execution is possible give sequence (Ans format : A, B, C) else give no safe sequence (Ans format :no) * ···/3



3 ✗

✗ From the following mix of jobs, job lengths, and arrival times, assume a time slice of 10 and compute the completion time of Job 1 (assuming FIFO scheduling algo) is * 0/2

Job	length	arrival time
0	75	0
1	40	10
2	25	10
3	20	80
4	45	85

130 ✗

✓ In ..... jobs are always put on the highest priority queue when they become ready to run * 1/1

- (●) MLFQ ✓
- ( ) SRT
- ( ) FIFO
- ( ) RR
- ( ) priority

✓ 5 jobs, of length 50, 40, 30, 20, and 10 seconds each, time slice 1 second, context switch time of 0 seconds. Assuming FCFS, average completion time is * 2/2

Job	Length
1	50
2	40
3	30
4	20
5	10

110 ✓

✗ A system has 6 identical resources and N processes competing for them. Each process can request atmost 2 resources. The minimum value of N that could lead to deadlock is * ···/2

6 ✗

✗ Consider the following mix of jobs, job lengths, and arrival times, assume a time slice of 10 and compute the completion time of Job 0 (assuming RR scheduling algo) is * 0/2

Job	length	arrival time
0	75	0
1	40	10
2	25	10
3	20	80
4	45	85

130 ✗

Find the best match *

	not require programmers to recheck conditions	'if' statement	“while” loop	signaling thread simply placed the signaled thread on the run queue and continues executing	signal() operation from one thread immediately wakes up a sleeping thread, hands the lock to the sleeping thread, and starts the sleeping thread executing;	Score	
Hoare scheduling	☑	☐	☐	☐	☐	0/1	✗
Mesa-scheduling	☑	☐	☐	☐	☐	0/1	✗
Mesa-monitors	☑	☐	☐	☐	☐	0/1	✗
Hoare-monitors	☑	☐	☐	☐	☐	0/1	✗

✗ Consider the following mix of jobs, job lengths, and arrival times, assume a time slice of 10 and compute the completion time of Job 0 (assuming SRTF scheduling algo) is * 0/2

Job	length	arrival time
0	75	0
1	40	10
2	25	10
3	20	80
4	45	85

75 ✗

✓ * 2/2

Semaphores provide a primitive yet powerful and flexible tool for enforcing mutual exclusion and for co-coordinating processes called .....

A) monitor
B) message passing
C) strong semaphore
D) binary semaphore

- (●) A ✓
- ( ) B
- ( ) C
- ( ) D

✗ *

0/2

Suppose that a process omits wait(mutex) or signal(mutex) or both. In this case:
A) Processes will starve to enter critical section
B) Either mutual exclusion is violated or deadlock will occur
C) Several Processes may be executing in their critical section
D) Processes will not starve to enter critical section

- ○ A
- ○ B
- ◉ C ✗
- ○ D

✓ *

1/1

In ..... only one process at a time is allowed into its critical section, among all processes that have critical sections for the same resource.
A) Synchronization
B) Mutual Exclusion
C) Shared Data
D) Race Condition

- ○ A
- ◉ B ✓
- ○ C
- ○ D

✓ The algorithmic approach to implementing critical sections did not employ indivisible instructions in a computer to avoid race conditions. * 2/2

- (●) True ✓
- ( ) False

Roll Number *

1901Cs65

✗ * 0/2

```
Which of the following facility or capacity are required to provide support for the mutual exclusion?

i) A process that halts in its noncritical section must do so without interfering with other processes.
ii) The assumption should be made about relative process speeds or the number of processors.
iii) A process remains inside its critical section for a finite time only

A) i and ii only
B) ii and iii only
C) i and iii only
D) All i, ii and iii
```

- ( ) A
- ( ) B
- ( ) C
- (●) D ✗

✓ 5 jobs, of length 50, 40, 30, 20, and 10 seconds each, time slice 1 second, context switch time of 0 seconds. Assuming SJF average completion time is * 2/2

Job	Length
1	50
2	40
3	30
4	20
5	10

70 ✓

✓ *

1/1

The Dining Philosophers Problem Solution is
A) Deadlock Free Solution
A) Starvation Free Solution
B) Page Fault Free Solution
C) All of the above

- ○ A
- ○ B
- ○ C
- ◉ D ✓

✓ *

1/1

The …… condition can be prevented by defining a linear ordering of resource types.
A) Mutual Exclusion
B) Hold and Wait
C) Preemption
D) Circular Wait

- ○ A
- ○ B
- ○ C
- ◉ D ✓

Match the following *

	discriminates against short jobs since any short jobs arriving after long jobs will have a longer waiting time	treats all jobs equally so short jobs will be able to leave the system faster since they will finish first	they do not discriminate favorably toward short jobs	they discriminate favorably toward short jobs	Do not treats all jobs equally so short jobs will be able to leave the system faster since they will finish first	Score	
FCFS	☐	☑	☑	☐	☑	0/1	✗
RR	☐	☐	☑	☐	☐	0/1	✗
Multilevel feedback queues	☐	☑	☐	☐	☐	0/1	✗

✗ *

0/1

Assume we have n threads at different priority levels and that they all use a lock l, which schedules waiting threads in FIFO order. Describe a plausible steady state behavior of this system.

- A. Lowest Priority thread grabs lock and gets prempted
- B. t Priority thread grabs lock and blocked
- C. System will run all other threads until they block on the queue
- D. low priority tD thread which will release the lock , until it reacquires the lock and then get places on the queue
- (●) A, B, C ✗
- A, B, C, and D

✗ Assuming one resource class. If C claims 9 instead of 7 . If safe sequence of execution is possible give sequence (Ans format : A, B, C) else give no safe sequence (Ans format : no) *

···/3

process	holding	max claims
A	4	6
B	4	11
C	2	7

unallocated: 2

3 ✗

✗ *

0/2

All processes share a semaphore variable **mutex**, initialized to 1. Each process must execute wait(mutex) before entering the critical section and signal(mutex) afterward.
Suppose a process executes in the following manner.

Signal (mutex)
.....
Critical section
.....
wait(mutex)

In this situation:
A) a deadlock will occur
B) processes will starve to enter critical section
C) several processes maybe executing in their critical section
D) all of the mentioned

- ○ A
- ◉ B ✗
- ○ C
- ○ D

✓ *

2/2

```
_____ when a process leaves a critical section and more than one process is waiting,
                                        the selection of a waiting process is arbitrary.


A) Busy waiting is employed
B) Starvation is possible
C) Deadlock is possible
D) All of the above
```

- ○ A
- ◉ B ✓
- ○ C
- ○ D

Name *

Tarusi Mitta;

✓ * 1/1

Which of the following is known as uninterruptible unit
A) Single
B) Atomic
C) Semaphores
D) Static

- ( ) A
- (•) B ✓
- ( ) C
- ( ) D

Match the following *

	Piece of code that only one thread can execute at once	Ensuring that only one thread does a particular thing at a time	Isolating program faults to an address space	Using atomic operations to ensure cooperation between threads	Score	
Synchronization	☐	☑	☐	☐	0/1	✗
Mutual exclusion	☑	☐	☐	☐	0/1	✗
Critical section	☐	☐	☐	☑	0/1	✗