2022_CS341-OperatingSystem_MSE

* Required

Email *

Your email

What is the content of the Matrix Need (give answer in the form (0, 1, 0, 0), (1, 7, 5, 0),(1,0, 1, 1), (0,2, 2, 0), (1, 2, 4, 2) * 2 points

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

Your answer

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. In RR the Turnaround time time for P2 is * 1 point

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

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 * 1 point

Your answer

Kernel-Level threads can be Created as many as needed * 1 point

- YES
- No
- Depends on OS

Request edit access

Mechanism used in modern computing systems to allow the OS to regain control of the CPU from a misbehaving application program (e.g. an application program in an infinite loop). * 2 points

Your answer

The operating system kernel is aware of the threads in the user space * 1 point

- ◯ True
- ◯ False

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. In FCFS the Waiting time time for P5 is * 1 point

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

Name *

Your answer

Request edit access

Give one of the possible outputs of the concurrent execution of process X and process Y , assuming output is not buffered (it is flushed immediately)? * 1 point

Process X	*Process Y*

`while (a==0)`
`  {do-nothing};`
`printf("4");`
`b=1;`
`b=0;`
`printf("A");`
	`printf("1");`
	`a=1;`
	`while (b==0)`
	`  {do-nothing};`
	`printf("3");`

Your answer

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. In RR the turnaround time time for P4 is * 1 point

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

Request edit access

When working with semaphores in a synchronization algorithm, the placement of signal (i.e. V) statements is important and could result in deadlock if done incorrectly. * 1 point

- ◯ True
- ◯ False

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. In SJF the Turnaround time time for P5 is * 2 points

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

Request edit access

*

1 point

The enter_CS() and leave_CS() functions to implement critical section of a process are realized using test-and-set instruction as follows:

```
void enter_CS(X)
{
        while (test-and-set(X)) ;
}
void leave_CS(X)
{
        X=0;
}
```

In the above solution, X is a memory location associated with the CS and is initialized to 0.

Now consider the following statements:

I. The above solution to CS problem is deadlock-free
II. The solution is starvation free.
III. The processes enter CS in FIFO order.
IV. More than one process can enter CS at the same time.

Which of the above statements are TRUE?

(A) I only

(B) I and II

(C) II and III

(D) IV only

- A
- B
- C
- D

Which of the following scheduling algorithms will not result in starvation? * 1 point

- First-Come First-Served (FCFS)
- Shortest Job First (SJF) (Non-preemptive)
- Shortest Remaining Time First (SRTF) (Preemptive)
- Round Robin (RR)
- Priority (Preemptive)

Request edit access

In UNIX, the exec() system call creates a new child process, and the fork system call replaces the current process's program with a new program and executes it. * 1 point

- ◯ True
- ◯ False
- ◯ Depends

A scheduler using the Round Robin algorithm is susceptible to the convoy effect * 1 point

- ◯ True
- ◯ False
- ◯ Depends

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. In FCFS the Turnaround time for P5 is * 1 point

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

Request edit access

When a process blocks on I/O, it is placed in the ready queue * 1 point

- True
- False
- Depends

* 1 point

Which of the following is NOT true of deadlock prevention and deadlock avoidance schemes?

(a) In deadlock prevention, the request for resources is always granted if the resulting state is safe
(b) In deadlock avoidance, the request for resources is always granted if the resulting state is safe
(c) Deadlock avoidance is less restrictive than deadlock prevention
(d) Deadlock avoidance requires knowledge of resource requirements a priori

- A
- B
- C
- D

With preemptive scheduling, the only time the scheduler considers running a different process is when the current one runs to completion. * 1 point

- True
- False
- Depends

Request edit access

Assuming that neither fork nor wait fail and all processes run to normal completion. Output of the program is (Ans. format:2201411) 4 points

```
main() {
   if (fork() == 0) {
      if (fork() == 0) {
         printf("3");
      }
      else {
         pid_t pid;
         int status;
         if ((pid = wait(&status)) > 0) {
            printf("4");
         }
      }
  }
  else {
     if (fork() == 0) {
        printf("1");
        exit(0);
     }
     printf("2");
  }
  printf("0");
  return 0;
}
```

Your answer

Request edit access

Which of the following operations require the executing code to be operating with high privilege? (A) Implementing a monitor(B) Performing a semaphore P operation(C) Accessing the device registers of an I/O device, e.g. the disk, keyboard, or network card(D) Disabling interrupts(E) Making a system call * 1 point

- A and E
- A and B
- C and D
- D and E
- Other:

*

3 points

Consider the following transactions with data items P and Q initialized to zero:

T_1 : read (P);

read (Q);

if P = 0 then Q : = Q+1;

write (Q)

T_2 : read (Q);

read (P);

if Q = 0 then P : = P+1;

write (P)

Any none-serial interleaving of T_1 and T_2 for concurrent execution leads to

(a) A serializable schedule
(b) A Schedule that is not conflict serializable
(c) A conflict serializable schedule
(d) A schedule for which a precedence graph cannot be drawn

- A
- B
- C
- D

Request edit access

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller pri ority number implies a higher priority), and RR (quantum = 1) scheduling. In FCFS the Turnaround time for P1 is * 1 point

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

A system call can be caused by events external to the CPU * 1 point

- ◯ True
- ◯ False

Request edit access

Is the system in a safe state? *

2 points

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

- ◯ Yes
- ◯ No

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. In nonpreemptive priority the Waiting time time time for P5 is *

1 point

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

Request edit access

Is the system in a safe state? * 1 point

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

- ◯ Yes
- ◯ No

User-Level threads can be created as many as needed * 1 point

- ◯ True
- ◯ False

stack and Registers shared by threads within a single process * 1 point

- ◯ True
- ◯ False

Multi-threading requires operating system support for managing multiple PCBs * 2 points

- ◯ True
- ◯ False

Request edit access

Peterson's Algorithm generalized to many threads is called ..... 1 point

*

- ◯ Bounded Waiting Algorithm
- ◯ Bakery Algorithm
- ◯ Multi-writer Algorithm
- ◯ precedence graphs Algorithm
- ◯ Other:

*

2 points

Consider a system having 'm' resources of the same type. These resources are shared by 3 processes A,B,C, which have peak time demands of 3,4,6 respectively. The maximum value of 'm' that ensures that deadlock will never occur is _____

Your answer

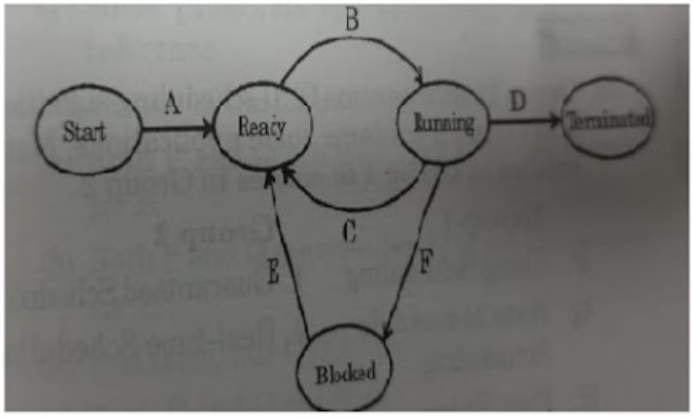
Find the best match *

5 points

	simplicity of construction, debugging, and extensible	monolithic performance with layered flexibility	protection/isolation	easy to extend, port. More reliable and secure	Better performance	Deadlo Free
Monolithic	☐	☐	☐	☐	☐	☐
Layered	☐	☐	☐	☐	☐	☐
Micro-kernel	☐	☐	☐	☐	☐	☐
Modular	☐	☐	☐	☐	☐	☐
Virtual Machines	☐	☐	☐	☐	☐	☐

2 points

In the following process state transition diagram for a uniprocessor system, assume that there are always some processes in the ready state:



Now consider the following statements:

(a) If a process makes a transition D, it would result in another process making transition A immediately.
(b) A process P2 in blocked state can make transition E while another process P1 is in running state.
(c) The OS uses preemptive scheduling.
(d) The OS uses non-preemptive scheduling.

- ◯ (a) is correct
- ◯ (b) and (d) are correct
- ◯ (b) and (c) are correct
- ◯ (d) and (c) are correct

Request edit access

*

2 points

The following program consists of 3 concurrent processes and 3 binary semaphores. The semaphores are initialized as

S0=1, S1=0, S2=0.

Process P0	Process P1	ProcessP2

while (true) {
wait (S0);
print '0';
release (S1);
release (S2);
}
	wait (S1);
	release (S0);
		wait (S2);
		release (S0);

How many times will process P0 print '0'?

(a) At least twice
(b) Exactly twice
(c) Exactly thrice
(d) Exactly once

- A
- B
- C
- D

Request edit access

If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately? (If yes give Process sequence (format :P0,P1,P3,P2,P4), otherwise give answer No) * 2 points

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

Your answer

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. In RR the Waiting time time time for P5 is * 1 point

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

Request edit access

*

1 point

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 _____

Your answer

When using interrupts, the CPU interrupts I/O devices when an I/O event happens *

1 point

- True
- False
- Depends

*

2 points

A process executes the following code

```
for ( i = 0; i < n; i++) for k ();
```

The total number of child processes created is

(a) n
(b) 2^n-1
(c) 2^n
(d) 2^{n+1}-1

- A
- B
- C
- D

Request edit access

At a particular time the value of a counting semaphone is 10. It will become 7 after * 1 point

- ◯ 3P
- ◯ 3V
- ◯ 3P and 1V
- ◯ 3V and 1P

Match the following (best match) * 4 points

	increment or block if already 0	increment and wake up process if any	decrement or block if already 0	Used for mutual exclusion	Used for synchronization
P(S)	☐	☐	☐	☐	☐
V(S)	☐	☐	☐	☐	☐
Counting Semaphores	☐	☐	☐	☐	☐
Binary Semaphores	☐	☐	☐	☐	☐

*

1 point

Which of the following statements are true?

I. Shortest remaining time first scheduling may cause starvation
II. Preemptive scheduling may cause starvation
III. Round robin is better than FCFS in terms of response time

(a) I only
(b) I and III only
(c) II and III only
(d) I, II and III

- A
- B
- C
- D

*

2 points

```
Which of the following is not an advantage about thread?

A. Threads minimize the context switching time.
B. Use of threads provides concurrency within a process.
C. kernel is single threaded
D. All of the above
```

- A
- B
- C
- D

Request edit access

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. In RR the Turnaround time time for P5 is * 1 point

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

* 1 point

In a time-sharing operating system, when the time slot given to a process is completed, the process goes from the RUNNING state to the

(a) BLOCKED state
(b) READY state
(c) SUSPENDED state
(d) TERMINATED state

- A
- B
- C
- D

Request edit access

User-level threads are threads that the OS is not aware of? * 1 point

- false
- True
- Depends

If a thread dies, its stack is reclaimed * 1 point

- True
- False

* 1 point

```
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

Heap and Code shared by threads within a single process * 1 point

- ◯ True
- ◯ False

The following processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0. Mr X tests the the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. In nonpreemptive priority the Turnaround time time time for P4 is * 1 point

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

Your answer

Roll Number *

Your answer

Request edit access

*

1 point

Consider the following statements about user level threads and kernel level threads.
Which one of the following statements is FALSE?

(A) Context switch time is longer for kernel level threads than for user level threads.

(B) User level threads do not need any hardware support.

(C) Related kernel level threads can be scheduled on different processors in a multi-processor system.

(D) Blocking one kernel level thread blocks all related threads.

- A
- B
- C
- D

Page 1 of 1

Submit Clear form