Type: MCQ

Q1. On a system using multilevel feedback queues, a CPU-bound process requires 40 seconds to execute. If the first queue uses a time quantum of 2 and at each level the time quantum increases by 5 time units how many times will the job be interrupted and on which queue will it be when it terminates? (0.5)

1. **4 interrupts, 5th queue
2. 5 interrupts, 6th queue
3. 3 interrupts, 4th queue
4. 4 interrupts, 4th queue

Q2. Counting semaphore (0.5)

1. Generalize the notion of a binary semaphore.
2. Are used for managing multiple instances of a resource.
3. Can use queuing to manage waiting processes.
4. **All of the above.

Q3. The key idea behind a _____is that sometimes it can be advantageous to remove a process from memory (and from active contention for the CPU) and thus reduce the degree of multiprogramming. (0.5)

1. Short term scheduler
2. **Medium-term scheduler
3. Long term scheduler
4. Both short and long term scheduler

Q4. On a system using non preemptive scheduling, processes with expected run times of 5, 18, 9 and 12 are in the ready queue. In what order should they be run to minimize wait time? (0.5)

1. 5, 18, 9 and 12
2. **5, 9, 12 and 18
3. 9, 5, 18 and 12
4. 18, 12, 9 and 5

Q5. _____ is a thread library available for Solaris systems, adopted in early versions of Java and used the many-to-one model. (0.5)

1. Pink Threads
2. Red Threads
3. **Green Threads
4. Black Threads

Q6. When several processes access the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place is called (0.5)

1. **Race condition
2. Progress
3. Critical Section
4. Mutual exclusion

Q7. Given a system with exponential averaging with $\alpha = 0.5$, what would be the next expected burst time for a process with burst times of 4, 7, 3 and 4, and an initial value for e_1 of 10? (0.5)

1. 4
2. 5.35
3. 6.5
4. **4.5

Q8. Consider a system consisting of 4 resources of the same type shared by 3 processes, each of which needs at most two resources. Which of the following is true? (0.5)

1. The system is not safe
2. **The system is deadlock-free
3. The system is in a deadlock.
4. Process undergoes starvation

Q9. Suppose the operating system is using semaphores to handle process synchronization. Let there be three processes that end up being blocked and placed in suspended list. Which of the following algorithms is used for waking the process. (0.5)

1. Round Robin
2. LIFO
3. **First Come First
4. Priority

Q10. Consider the system consisting of four processes and a single resource. The current state of the system is given here.

Process	Max	Allocation
P0	3	1
P1	2	1
P2	9	3
P3	7	2

For this state to be safe, what should be the minimum number of instances of this resource? (0.5)

1. 9

2. 8
3. 11
4. **10

Type: DES

Q11. Explain the responsibilities of the operating system in connection with: a. Process Management b. Memory Management. Explain the difference between Multitasking and Multiprogramming . (4)

Ans: . **Scheme: Each valid point 0.5 M**

A. List and explain the responsibilities of the operating system in connection with: 2M

a. Process management
b. Memory management

A. The operating system is responsible for the following activities in connection with process management:

- Scheduling processes and threads on the CPUs
- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication

The operating system is responsible for the following activities in connection with memory management:

- Keeping track of which parts of memory are currently being used and who is using them
- Deciding which processes (or parts of processes) and data to move into and out of memory
- Allocating and deallocating memory space as needed

B. Explain the difference between Multitasking and Multiprogramming.

Differences between multitasking and multi-programming are:

Multi programming:

- **Multiprogramming** increases CPUutilization by organizing jobs (code and data) so that the CPU always has one to execute.
- In a multiprogrammed system, the operating system simply switches to, and executes, another job. When ***that*** job needs to wait, the CPU switches to ***another*** job, and so on.

Multi tasking (Time shared):

- In time-sharing systems, the CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.
- A time-shared operating system allows many users to share the computer simultaneously.
- Time sharing requires an **interactive** computer system, which provides direct communication between the user and the system.

Q12. Elaborate blocking and non-blocking in process synchronization. Outline the structure of producer and consumer process. (4)

Ans:

Blocking is considered **synchronous**

- **Blocking send** -- the sender is blocked until the message is received
- **Blocking receive** -- the receiver is blocked until a message is available

Non-blocking is considered **asynchronous**

- **Non-blocking send** -- the sender sends the message and continue
- **Non-blocking receive** -- the receiver receives:
  - A valid message, or
  - Null message

Different combinations possible

If both send and receive are blocking, we have a **rendezvous**

2M

```
item next_consumed;

while (true) {
     while (in == out)
        ; /* do nothing */

     next_consumed = buffer[out];
     out = (out + 1) % BUFFER_SIZE;

     /* consume the item in next_consumed */
}
```

The consumer process using shared memory.

1M

```
item next_produced;

while (true) {
     /* produce an item in next_produced */

     while (((in + 1) % BUFFER_SIZE) == out)
        ; /* do nothing */

     buffer[in] = next_produced;
     in = (in + 1) % BUFFER_SIZE;
}
```

The producer process using shared memory.

1M

Q13. Operating systems has two services namely user services and system services. Elaborate functions of each services . (3)

1. Ans: **System Services:**
    - **Hardware Abstraction:** The operating system abstracts the underlying hardware, providing a consistent interface for software. This abstraction shields software from hardware details, ensuring compatibility across different hardware platforms.
    - **Resource Management:** It manages system resources like CPU time, memory, and storage. This includes task scheduling to ensure fair resource allocation and efficient utilization.
    - **File System Management:** The OS provides file-related services, enabling applications to create, read, write, and delete files. It maintains file organization and access permissions.
    - **Device Management:** It controls input and output devices, handling communication between software and hardware components. This ensures that applications can interact with various devices without needing to know their specifics.
    - **Security and Access Control:** The OS enforces security policies, manages user accounts, and regulates access to system resources to protect against unauthorized access.
    - **Error Handling:** It monitors and responds to hardware and software errors, minimizing disruptions and providing error messages for diagnosis.
2. **User Services:**
    - **User Interface:** The OS offers interfaces for users to interact with the computer, including graphical user interfaces (GUIs) and command-line interfaces (CLIs).

- **Application Support:** It provides software libraries and APIs that help developers create applications. This includes system calls and libraries for common tasks.
- **Utilities:** The OS includes utility programs for system maintenance, such as disk cleanup, backup, and performance monitoring.
- **Networking:** It manages network connections, configurations, and protocols, enabling communication between devices and network services.
- **Multitasking:** The OS allows multiple applications to run simultaneously, switching between them and managing their execution.
- **User Management:** It facilitates user account creation, permissions management, and user customization.
- **Accessibility:** The OS includes features and tools to make computing accessible to individuals with disabilities, such as screen readers and keyboard shortcuts.

in summary, system services focus on low-level management, resource allocation, and hardware control, ensuring the efficient operation of the computer. User services are more user-centric, providing a user-friendly environment, tools for application development, and interfaces for interacting with the system. These two categories work together to create a functional and user-friendly computing experience.

Services :1M minimum 4

Functions:1M minimum 4

discuss how they differ-1M

Q14. Consider the following snapshot of a system:

	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	

a. Demonstrate the process sequence if it is in a safe state.
b. If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately? Justify your answer.
(Note: Show the intermediate steps.). (3)

Ans:

Scheme:

Need matrix – 0.5 M+0.5 M in a and b

Safe sequence – 0.5 M+0.5 M in a and b

Intermediate steps – 1M+1M in a and b

Scheme for deadlocks

(a)

Need matrix

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

Available data

A	B	C	D	
1	5	2	0	Initially
1	5	3	2	after P_0
		P_1 is waiting		
2	8	8	6	after P_2
12	14	11	8	after P_3
2	14	12	12	after P_4

Safe sequence is $P_0 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$

Available resource at the end →

3 14 12 12

(b) P_1 request for resource (0, 4, 2, 0)

	Allocation				Max				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D				
P_0	0	0	1	2	0	0	1	2	0	0	0	0	1	1	0	0
P_1	1	4	2	0	1	7	5	0	0	3	3	0	1	1	1	2
P_2	1	3	5	4	2	3	5	6	1	0	0	2	2	4	6	6
P_3	0	6	3	2	0	6	5	2	0	0	2	0	2	10	9	8
P_4	0	0	1	4	0	6	5	6	0	6	4	2	2	10	10	12

$P_0 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$

Available resource at the end →

3 14 12 12.

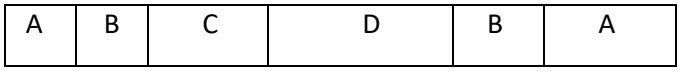
Yes request can be granted.

Q15. For the processes listed below, draw a chart illustrating their execution using preemptive and non-preemptive priority scheduling. A larger priority number has higher priority. Calculate the average TAT and WT.

Process	Arrival Time ms	Burst in ms	Priority
A	0	4	3
B	1	3	4
C	2	3	6
D	3	5	5

. (3)

1. Ans: Preemptive



0 1 2 5 10 12 15

0.5M

TAT: A = 15, B = 11, C =3, D = 7 avg = 9ms 0.5M

WT: A = 11, B = 8, C= 0, D = 2 avg = 5.25ms 0.5M

B. Non-Preemptive

0 4 7 12 15

0.5M

TAT: A = 4, B = 14, C =5, D = 9 avg = 8ms 0.5M

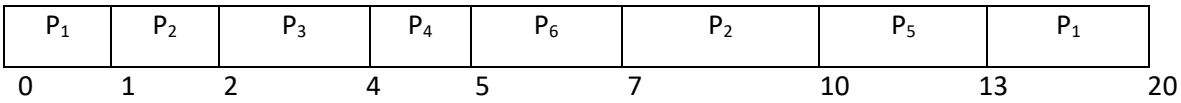
WT: A = 0, B = 11, C= 2, D = 4 avg = 4.25ms 0.5M

Q16. For the processes given below, draw a gantt chart and calculate average TAT, average WT and response time for each process using preemptive shortest job first.

Process	Arrival Time ms	Burst in ms
P_1	0	8
P_2	1	4
P_3	2	2
P_4	3	1
P_5	4	3
P_6	5	2

. (3)

Ans:



1M (0.5M + 0.5M FCFS)

TAT: $P_1 = 20$, $P_2 = 9$, $P_3 = 2$, $P_4 = 2$, $P_5 = 9$, $P_6 = 2$ avg TAT = 7.33ms 0.5M

WT: $P_1 = 12$, $P_2 = 5$, $P_3 = 0$, $P_4 = 1$, $P_5 = 6$, $P_6 = 0$ avg WT =4ms 0.5M

RT: $P_1 = 0$, $P_2 = 0$, $P_3 = 0$, $P_4 = 1$, $P_5 = 6$, $P_6 = 0$ 1M

Q17. An application using threads is to be developed for implementation on single CPU computer systems as well as multiple-CPU computer systems.

i. Explain what thread model you would use for implementing it on a single-CPU computer system. Give detailed justifications and mention whether the OS should provide any factility in addition to the model of threads chosen by you.

ii. Explain what thread model you would use for implementing it on a multiple-CPU computer system. Give detailed justifications and mention whether the OS should provide any facility in addition to the model of threads chosen by you. (3)

Ans: (i) Parallelism of the threads is not important; only concurrency is important- 0.5 Mark

Low thread switching overhead is important Because switching is performed by the thread library, thread switching overhead is low. - 0.5 Mark

Hence use user-level threads A library to perform certain actions without blocking a thread is essential-0.5 Marks

(ii) Parallelism between threads is essential 0.5 Mark

Hence kernel-level threads 0.5 Mark

Thread switching overhead is not so important .No special provisions needed in the OS 0.5 Mark

Q18. Elaborate on the advantages and disadvantages of spinlock. (2)

Ans: The main disadvantage of the implementation given here is that it requires busy waiting. While a process is in its critical section, any other process that tries to enter its critical section must loop continuously in the call to acquire(). In fact, this type of mutex lock is also called a spinlock because the process “spins” while waiting for the lock to become available. (We see the same issue with the code examples illustrating the test and set()

instruction and the compare and swap() instruction.) This continual looping is clearly a problem in a real multiprogramming system, where a single CPU is shared among many processes. Busy waiting wastes CPU cycles that some other process might be able to use productively. -1 mark

Spinlocks do have an advantage, however, in that no context switch is required when a process must wait on a lock, and a context switch may take considerable time. Thus, when locks are expected to be held for short times, spinlocks are useful. They are often employed onmultiprocessor systems where one thread can “spin” on one processor while another thread performs its critical section on another processor. -1 mark