USN | | | | | | | | | | |

Internal Assessment Test 1 – November 2024

Sub:	**Operating Systems**					Sub Code:	BCS303	Branch:	ISE	
Date:	**11/11/2024**	Duration:	90 min's	Max Marks:	50	Sem/Sec:	III A, B & C		OBE	

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1.a	Differentiate between multiprogramming, multiprocessing and multitasking systems.	6	CO1	L1
1.b	Explain about any two operating systems that are currently dominating mobile computing	4	CO1	L2
2	Explain dual mode operation in Operating System and illustrate the transition between them with neat block diagram.	10	CO1	L2
3	Explain process states with state transition diagram. Also explain PCB with a neat diagram	10	CO2	L2
4		10	CO2	L3

Process	Arrival Time	Burst Time	Priority
P1	4	9	3
P2	3	7	1
P3	0	5	0
P4	2	4	2

For each scheduling algorithm, FCFS, SJF (non-preemptive) , Priority (smaller priority number implies lower scheduling priority) and RR (quantum = 3) , do the following

a. Draw a Gantt Chart b. Calculate Average Turn Around Time and Average Waiting Time

Determine which algorithm performs better in this case.

5 a	Explain the various models of Multi-threading with one Pros and Cons.	6	CO2	L2
5 b	Differentiate between Message Passing and Shared Memory Systems	4	CO2	L1
6 a	Demonstrate the process synchronization issues in Producer Consumer problem using pseudocode.	6	CO3	L3
6 b	Explain all necessary conditions to ensure the solution to Critical Section Problem.	4	CO3	L2

C I C C I HOD

USN | | | | | | | | | | |

Internal Assessment Test 1 – November 2024

Sub:	**Operating Systems**					Sub Code:	BCS303	Branch:	ISE	
Date:	**/11/2024**	Duration:	90 min's	Max Marks:	50	Sem/Sec:	III A, B & C		OBE	

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1.a	Differentiate between multiprogramming, multiprocessing and multitasking systems.	6	CO1	L1
1.b	Explain about any two operating systems that are currently dominating mobile computing	4	CO1	L2
2	Explain dual mode operation in Operating System and illustrate the transition between them with neat block diagram.	10	CO1	L2
3	Explain process states with state transition diagram. Also explain PCB with a neat diagram	10	CO2	L2
4		10	CO2	L3

Process	Arrival Time	Burst Time	Priority
P1	4	9	3
P2	3	7	1
P3	0	5	0
P4	2	4	2

For each scheduling algorithm, FCFS, SJF (non-preemptive) , Priority (smaller priority number implies lower scheduling priority) and RR (quantum = 3) , do the following

a. Draw a Gantt Chart

b. Calculate Average Turn Around Time and Average Waiting Time

Determine which algorithm performs better in this case.

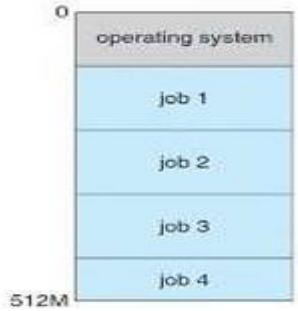
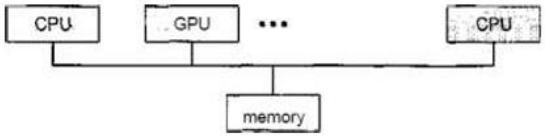
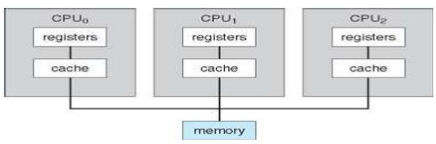
5 a	Explain the various models of Multi-threading with one Pros and Cons.	6	CO2	L2
5 b	Differentiate between Message Passing and Shared Memory Systems	4	CO2	L1
6 a	Demonstrate the process synchronization issues in Producer Consumer problem using pseudocode.	6	CO3	L3
6 b	Explain all necessary conditions to ensure the solution to Critical Section Problem.	4	CO3	L2

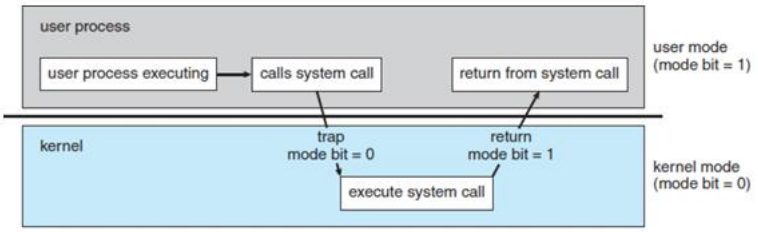
C I C C I HOD

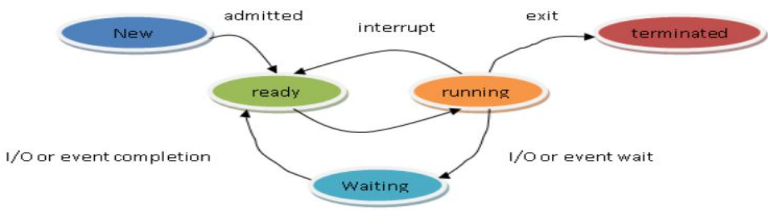
USN | | | | | | | | | | |

Internal Assessment Test 1 – November 2024

Sub:	**Operating Systems**					Sub Code:	BCS303	Branch:	ISE
Date:	**11/11/2024**	Duration:	90 min's	Max Marks:	50	Sem/Sec:	III A, B & C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1.a	Differentiate between multiprogramming, multiprocessing and multitasking systems. **Solution** **Multiprogramming**: A single user cannot keep either the CPU or the I/O devices busy at all times. Multiprogramming increases CPU utilization by organizing jobs, so that the CPU always has one to execute.  Fig - Memory layout for a multiprogramming system **Multitasking** Systems • In Time sharing (or multitasking) systems, a single 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. The user feels that all the programs are being executed at the same time.  **Multiprocessing** Systems 1) Asymmetric multiprocessing – (Master/Slave architecture) Here each processor is assigned a specific task, by the master processor. A master processor controls the other processors in the system. It schedules and allocates work to the slave processors. 2) Symmetric multiprocessing (SMP) – All the processors are considered peers. There is no master-slave relationship. All the processors have their own registers and CPU, only memory is shared.  3x2M each	6	CO1	L1
1.b	Explain about any two operating systems that are currently dominating mobile computing **Solution**	4	CO1	L2

	Android Operating System Android is an open-source mobile OS developed by Google and launched in 2008. Android is a Linux-based OS that uses Linux 2.6 to provide core services such as security, memory management, process management, network stack, and a driver model. It offers a wide range of libraries that enable the app developers to build different applications. Android applications are usually written in Java programming language. **Apple iOS** Apple iOS is a closed-source code mobile phone OS developed by Apple in 2007; it is used by Apple-only products (iPhone, iPod, and iPad). The iOS architecture is based on three layers incorporated with each other. Cocoa touch is a layer that provides some basic infrastructure used by applications. The second layer is the media layer, which provides audio services, animation video, image formats, and documents in addition to providing two-dimensional (2D) and 3D drawings and audio and video support. The third layer is the core OS, which provides core services such as low-level data types, start-up services, network connection, and access 2x2M each			
2	Explain dual mode operation in Operating System and illustrate the transition between them with neat block diagram. **Solution**: Since the operating system and the user programs share the hardware and software resources of the computer system, it has to be made sure that an error in a user program cannot cause problems to other programs and the Operating System running in the system. The approach taken is to use a hardware support that allows us to differentiate among various modes of execution. The system can be assumed to work in two separate modes of operation: 1. **User mode** 2. **Kernel mode** (supervisor mode, system mode, or privileged mode). A hardware bit of the computer, called the mode bit, is used to indicate the current mode: kernel (0) or user (1). With the mode bit, we are able to distinguish between a task that is executed by the operating system and one that is executed by the user. • When the computer system is executing a user application, the system is in user mode. When a user application requests a service from the operating system (via a system call), the transition from user to kernel mode takes place.  **Figure** Transition from user to kernel mode. 5M: Diagram 5M explanation	10	CO1	L2
3	Explain process states with state transition diagram. Also explain PCB with a neat diagram **Solution** A process is a program which is currently in execution.	10	CO2	L2

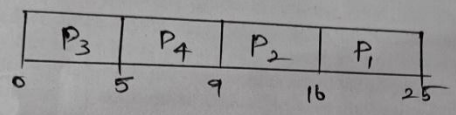
	A process also includes the process stack, which contains temporary data (such as local variables, function parameters, return address), and a data section, which contains global variables and a heap-memory allocated to a process to run and process state that defines its current state. A process changes its state during its execution. Each process may be in one of the following states: 1. New: when a new process is being created. 2. Running: A process is said to be in running state when instructions are being executed. 3. Waiting: The process is waiting for some event to occur (such as an I/O opn.). 4. Ready: The process is waiting for processor. 5. Terminated: The process has finished execution.  Process State Diagram Process Control Block (PCB): Operating system maintains one special data structure called Process Control Block (PCB). All the information about each process is stored in the process control block (PCB) which is maintained by operating system. It contains following information associated with a specific process. • Process state: It represents current status of the process. It may be new, ready, running or waiting. • Program counter: It indicates the address of the next instruction to be executed for this process. • CPU Registers: They include index registers, stack pointer and general purpose registers. It is used to save process state when an interrupt occurs, so that it can resume from that state. • CPU-scheduling information: it includes process priority, pointer to scheduling queue. • Memory management information: value of the base and limit registers, page tables depending on the memory system. • Accounting information: it contains an amount of CPU and real time used, time limits process number and so on. • I/O status information: It includes a list of I/O devices allocated to the process, a list of open files and so on. • Normally, operating system stores PCBs of processes into the ready queue for the process scheduling instead of the process itself. 2M Diagram 4M Explanation on state diagram 2M PCB contents listing 2M explanation on PCB			
4	FCFS, SJF, RR, Priority **Solution**	10	CO2	L1

4 FCFS, SJF, RR, Priority — 10 — CO2 — L1

Solution

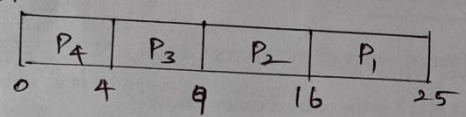
Process	A.T	B.T	Priority
P_1	4	9	3
P_2	3	7	1
P_3	0	5	0
P_4	2	4	2

FCFS



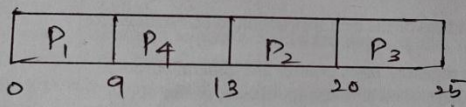
A.W.T = 5.25
A.TAT = 11.25

SJF



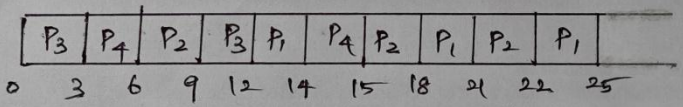
A.WT = 5.25
A.TAT = 11.5

PRIORITY:



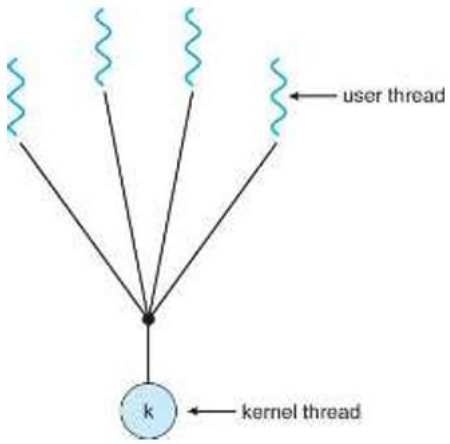
A.W.T = 10.5
A.TAT = 16.75

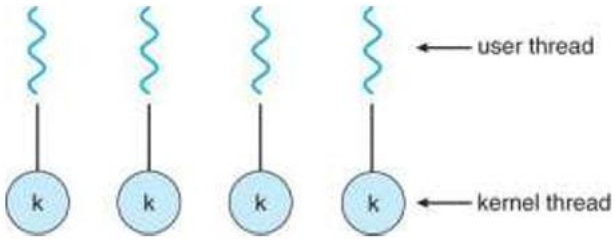
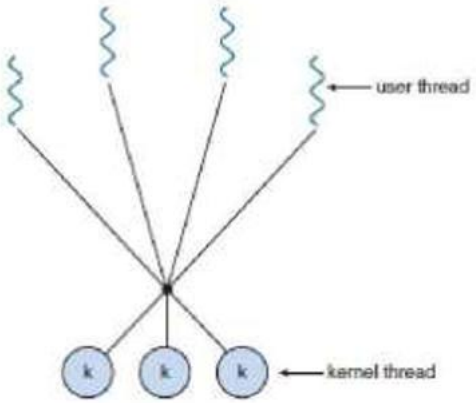
ROUND ROBIN:



A.W.T = 9.75
A.TAT = 16

FCFS & SJF algorithm performs well in this case.

5 a	Explain the various models of Multi threading with one Pros and Cons. **Solution**: Various models of Multi threading: 1. Many-to-one model 2. One-to-one mode and 3. Many-to-many model. **Many-to-One Model** • Many user-level threads are mapped to one kernel thread. Advantages: ▪ Thread management is done by the thread library in user space, so it is efficient. Disadvantages: ▪ The entire process will block if a thread makes a blocking system-call. ▪ Multiple threads are unable to run in parallel on multiprocessors. • For example: ▪ Solaris green threads ▪ GNU portable threads.  Fig: Many-to-one model **One-to-One Model** • Each user thread is mapped to a kernel thread. Advantages:	6	CO2	L2

	▪ It provides more concurrency by allowing another thread to run when a thread makes a blocking system-call. ▪ Multiple threads can run in parallel on multiprocessors. Disadvantage: ▪ Creating a user thread requires creating the corresponding kernel thread. • For example: ▪ Windows NT/XP/2000, Linux  Fig: one-to-one model **Many-to-Many Model** • Many user-level threads are multiplexed to a smaller number of kernel threads. Advantages: ▪ Developers can create as many user threads as necessary ▪ The kernel threads can run in parallel on a multiprocessor. ▪ When a thread performs a blocking system-call, kernel can schedule another thread for execution.  Fig: Many-to-many model 3x2M each			
5 b	Differentiate between Message Passing and Shared Memory Systems Solution	4	CO2	L1

Sl No	Shared Memory	Message passing
1.	A region of memory is shared by communicating processes, into which the information is written and read	Message exchange is done among the processes by using objects.
2.	Useful for sending large block of data	Useful for sending small data.
3.	System call is used only to create shared memory	System call is used during every read and write operation.
4.	Message is sent faster, as there are no system calls	Message is communicated slowly.

6 a	Solve the process synchronization issues in Producer Consumer problem with an	6	CO3	L2

example using relevant pseudocodes.

Solution

Producer-Consumer Example Using Shared Memory

• This is a classic example, in which one process is producing data and another process is consuming the data.

• The data is passed via an intermediary buffer (shared memory). The producer puts the data to the buffer and the consumer takes out the data from the buffer. A producer can produce one item while the consumer is consuming another item. The producer and consumer must be synchronized, so that the consumer does not try to consume an item that has not yet been produced. In this situation, the consumer must wait until an item is produced.

• There are two types of buffers into which information can be put –

• Unbounded buffer

• Bounded buffer

• With Unbounded buffer, there is no limit on the size of the buffer, and so on the data produced by producer. But the consumer may have to wait for new items.

• With bounded-buffer – As the buffer size is fixed. The producer has to wait if the buffer is full and the consumer has to wait if the buffer is empty.

The producer process –

Note that the buffer is full when [(in+1) % BUFFER_SIZE == out]

```
item nextProduced;

while (true) {
     /* produce an item in nextProduced */
     while ((in + 1) % BUFFER-SIZE) == out)
        ; /* do nothing */
     buffer[in] = nextProduced;
     in = (in + 1) % BUFFER_SIZE;
}
```

Figure The producer process.

The consumer process –

Note that the buffer is empty when [in == out]

```
item nextConsumed;

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

     nextConsumed = buffer[out];
     out = (out + 1) % BUFFER_SIZE;
     /* consume the item in nextConsumed */
}
```

Figure The consumer process.

Pseudo Code: 3M

Explanation: 3M

6 b	Explain all necessary conditions to ensure the solution to Critical Section Problem. **Solution**: A solution to the critical-section problem must satisfy the following three requirements:	4	CO3	L3

	1. Mutual exclusion: If process Pi is executing in its critical section, then no other processes can be executing in their critical sections. 2. Progress: If no process is executing in its critical section and some processes wish to enter their critical sections, then only those processes that are not executing in their remainder sections can participate in deciding which will enter its critical section next, and this selection cannot be postponed indefinitely. 3. Bounded waiting: There exists a bound, or limit, on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted. Explanation: 4M			