4022

No of Pages : 5 **Course Code : 20XW45**

Roll No:
(To be filled in by the candidate)

PSG COLLEGE OF TECHNOLOGY, COIMBATORE - 641 004

SEMESTER EXAMINATIONS, APRIL 2024

MSc - SOFTWARE SYSTEMS Semester : 4

20XW45 OPERATING SYSTEMS

Time : 3 Hours **Maximum Marks :100**

INSTRUCTIONS:

1. Answer **ALL** questions. Each question carries 20 Marks.
2. Subdivisions (a)(i) and (a)(ii) carries 2 marks each, subdivision (b) carries 6 marks each and subdivision (c) carries 10 marks each.
3. Course Outcome : Table

Qn.1	CO.1	Qn.2	CO.2	Qn.3	CO.3	Qn.4	CO.4	Qn.5	CO.5

COs	At the end of learning this course, students will be able to
CO1	Understand the operations managed by the OS, evolution of os and design strategies: micro-kernel vs monolithic
CO2	Execute and implement multiprocessing and multithreaded applications using various concurrency and messaging mechanisms
CO3	Develop a deep understanding of the concepts of cooperating processes, including communication, synchronization, and deadlock
CO4	Understand Memory Management functionalities implemented by OS, and the various partition schemes. Execute memory profiling to various programming examples
CO5	Critically evaluate the design of file system and I/O managers in various OS case studies and interface with them from higher level applications.

Question **(BTL)**

1. a) i) What is Dual Mode operation? For each of the following cases, write if a mode switch occurs. (L1)
   - A user program triggers a page fault.
   - The timer interrupt interrupts a user program.
   - A bug in the kernel's process scheduling algorithm causes a memory segmentation fault.
   - A user process sends a signal to another user process.

   ii) Consider the following C program. Assume there are no syntax errors and the program executes correctly. Assume the fork system calls succeed. What is the output printed to the screen when we execute the below program? (L2)

```
void main() {
        for(int i = 0; i < 4; i++) {
                int ret = fork();
                if(ret == 0)
                        printf("child %d\n", i);
        }
}
```

b) Explain 5-state model. For each of the following thread state transitions, (L3) say whether the transition is legal and how the transition occurs or why it cannot.

- Change from thread state BLOCKED to thread state RUNNING
- Change from thread state RUNNING to thread state BLOCKED
- Change from thread state RUNNABLE to thread state BLOCKED

c) Explain how operating systems have evolved over the years and provide three reasons for this evolution. (L1)

2. a) i) How the mixture of I/O bound processes with CPU bound processes maximizes system utilization? (L2)

ii) Inter-Process Communication (IPC) requires that two processes on the same host can somehow share information directly. Why does this require special support in the operating system? Compare the following IPC mechanisms in terms of how easily they support interaction between processes: pipes, shared memory, message passing. (L3)

b) Consider two threads A and B that perform two operations each. Let the operations of thread A be A1 and A2; let the operations of thread B be B1 and B2. We require that threads A and B each perform their first operation before either can proceed to the second operation. That is, we require that A1 be run before B2 and B1 before A2. Consider the following solutions based on semaphores for this problem (the code run by threads A and B is shown in two columns next to each other). For each solution, explain whether the solution is correct or not. If it is incorrect, you must also point out why the solution is incorrect. (L3)

```
/* Solution 1 */
sem A1Done = 0; sem B1Done = 0;
//Thread A                  //Thread B
A1                          B1
wait (B1Done)               wait (A1Done)
signal (A1Done)             signal (B1Done)
A2                          B2
```

```
/* Solution 2 */
sem A1Done = 0; sem B1Done = 0;
//Thread A                  //Thread B
A1                          B1
wait (B1Done)               signal (B1Done)
signal (A1Done)             wait (A1Done)
A2                          B2
```

```
/* Solution 3 */
sem A1Done = 0; sem B1Done = 0;
//Thread A                  //Thread B
A1                          B1
signal (A1Done)             signal (B1Done)
wait (B1Done)               wait (A1Done)
A2                          B2
```

c) Suppose the following processes arrive at the times indicated. The total available memory is 35 units. OS occupies 10 units and the remaining 25 units for user process. Assume all processes are CPU bound and the content switch time is 1. (L5)

Process	Arrival Time	CPU Burst Time	Memory Required (Units)
P1	0	5	3
P2	3	3	5
P3	7	9	8
P4	12	10	12
P5	18	16	2
P6	25	2	6
P7	29	8	9

- Calculate the turnaround time for each process using FCFS scheduling policy.
- Draw the memory map if First-fit allocation policy is used and processes are scheduled using FCFS.
- Draw the memory map if Best-fit allocation policy is used and processes are scheduled using FCFS.

3. a) i) Which of the following events requires the OS to update the page table pointer in the MMU (and flush the changes to the TLB)? Answer "update" or "no update". (L2)

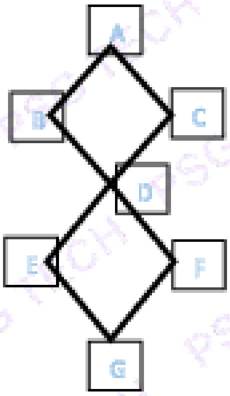
- A process moves from user mode to kernel mode.
- The OS switches context from one process to another.

ii) Fix the following code to avoid the possible deadlock. What condition did you remove by making your change? (L4)

Process A	Process B

acquire(L1)
acquire(L2)
release(L2)
release(L1)
	acquire(L2)
	acquire(L1)
	release(L1)
	release(L2)

b) Semaphores can be used to express scheduling constraints between activities performed by different threads. Consider the following flow diagram denoting the dependencies between activities A, B, C, D, E, F and G which are executed by 7 different threads. Write a Pseudo Code to implement these 7 threads to ensure these constraints using semaphore. (L3)



c) What is Deadlock? What are the necessary conditions for deadlock to occur? According to the Banker's algorithm, is the following system state safe? Why or Why not? Show your computation step-by-step and your safe sequence explicitly. (L4)

Claim / Max Matrix

	Camera	Printer	Bluetooth
Process A	2	1	2
Process B	5	5	3
Process C	5	4	4
Process D	1	0	2

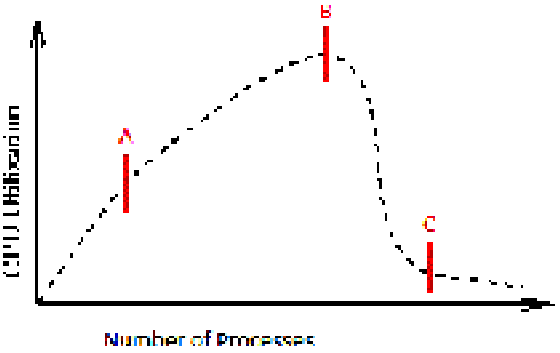
Current Allocation Matrix

	Camera	Printer	Bluetooth
Process A	1	1	0
Process B	2	4	1
Process C	4	2	3
Process D	1	0	1

Total Resources Available in the System :

Camera	Printer	Bluetooth
8	7	6

4. a) i) Consider the classic thrashing curve below. Note that on this curve three points are marked: "A", "B", and "C". (L4)



Assume that the computer is operating at point "B", so is productively busy. What would be the behavior of the computer, either it will stay near "B" or move much closer to "A" or "C" when,

- the size of the working set parameter, window size ∇ is increased
- the size of the working set parameter, Window size ∇ is decreased

Justify your answer.

ii) What is the impact of program structure and data structure in Locality of Reference and Page Fault Rate? (L2)

b) Assume there is an initial 1024 KB segment and memory is allocated using the Buddy system. Draw the tree illustrating how the following memory requests are allocated: (L3)

- Request 240 bytes
- Request 120 bytes
- Request 60 bytes
- Request 130 bytes.

c) Consider a simple system running a single process. The size of physical frames and logical pages is 16 bytes. The RAM can hold 3 physical frames. The virtual addresses of the process are 6 bits in size. The program generates the following 20 virtual address references as it runs on the CPU: 0, 1, 20, 2, 20, 21, 32, 31, 0, 60, 0, 0, 16, 1, 17, 18, (L4)

32, 31, 0, 61.

(Note: the 6-bit addresses are shown in decimal here.) Assume that the physical frames in RAM are initially empty and do not map to any logical page.

- Translate the virtual addresses above to logical page numbers referenced by the process. That is, write down the reference string of 20 page numbers corresponding to the virtual address accesses above.
- Calculate the number of page faults generated by the accesses above, assuming a FIFO page replacement algorithm. You must also correctly point out which page accesses in the reference string shown by you for the above question are responsible for the page faults.
- Repeat above for the LRU page replacement algorithm.
- What would be the lowest number of page faults achievable in this example, assuming an optimal page replacement algorithm were to be used? Repeat above for the optimal algorithm.

5. a) i) The following are some characteristics of disk allocation methods. Name three methods of disk allocation and indicate which characteristics they possess. (L2)

- Space wasted in index block
- Inefficient for random access
- Single file typically resides on a single cylinder
- Suffers external fragmentation
- Suffers internal fragmentation if initial file size allocation is too big.
- Unreliable because single error can cause loss of many blocks of data

ii) Consider the process of opening a new file that does not exist (obviously, creating it during opening), via the "open" system call. Describe changes to all the in-memory and disk-based file system structures (e.g., file tables, inodes, and directories) that occur as part of this system call implementation. Write clearly, listing the structure that is changed, and the change made to it. (L3)

b) What is a file? What are the various file operations? What are the different accessing methods for a file? (L1)

c) Disk requests come into the disk driver for cylinders 10, 22, 20, 2, 40, 6, and 38, in that order. Assume that the disk has 100 cylinders. A seek takes 6msec per cylinder moved. Compute the average seek time for the request sequence given above for (L4)

- First-come, First-served
- Shortest Seek Time First (SSTF)
- LOOK (with the disk-arm initially moving towards higher number cylinders from lower number cylinders)
- C-SCAN

In all the cases, the arm is initially at cylinder 20.

CO : COURSE OUTCOME; BTL : BLOOM'S TAXONOMY LEVEL

/END/

FD/JU