NATIONAL INSTITUTE OF TECHNOLOGY KURUKSHETRA

THEORY EXAMINATION

Question paper

Roll No: _____

Month and Year of the Examination: May/June, 2021 Branch: Computer Science

Subject: Operating System Subject Code:ITPC-20

Course: B.Tech Semester: IV

Time Duration: Two (2) Hours Maximum Marks:50

Total No. of Questions: Six (6)

Email: priyankaahlawat@nitkkr.ac.in, bharatisinha@nitkkr.ac.in Mobile:9034996998, 9729063343

Number of Questions to be Attempted: Five (5)

Note:

I. Attempt all parts of question together at one place. Marks allotted for each question are shown on the right hand margin. Write your answer with question no.

II. The Candidates, before starting to write the solutions, should please check the question paper for any discrepancy, and also

ensure that they have been delivered the question paper of right course no. and right subject title.

III. Unless stated otherwise, the symbols have their usual meanings in context with the subject. Assume suitably and state, additional data required, if any.

Q 1 (a) Write about the different state transitions in life cycle of a process and at least one system call associated with each transition. (2 marks)

(b) Which scheduling algorithm is best for periodic tasks and why? (2 marks)

(c) Consider the following three processes that arrive in a system at the specified times, along with the duration of their CPU bursts. Process P1 arrives at time t=0, and has a CPU burst of 10 time units. P2 arrives at t=2, and has a CPU burst of 2 units. P3 arrives at t=3, and has a CPU burst of 3 units. Assume that the processes execute only once for the duration of their CPU burst, and terminate immediately. Draw the gantt chart and calculate the time of completion of the three processes under each of the following scheduling policies. For each policy, you must state the completion time of all three processes, P1, P2, and P3. Assume there are no other processes in the scheduler's queue. For the preemptive policies, assume that a running process can be immediately preempted as soon as the new process arrives (if the policy should decide to premempt).(6 marks)

(i) First Come First Serve

(ii) Shortest Job First (non-preemptive)

(iii) Shortest Remaining Time First (preemptive)

(iv) Round robin (preemptive) with a time slice of (atmost) 5 units per process

(v) priority(preemptive)

(vi) priority(non-preemptive)

Q2 (a) Explain the disastrous consequences of race condition in banking transactions if synchronization conditions are

not met. Which classical synchronization problem is the credit and debit module similar to and how?(4 marks)

(b)(4 marks)

	Allocation				Max				Request				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P1	1	0	1	2	1	0	2	2	0	0	1	0	2	1	3	4
P2	0	1	0	1	0	2	1	2	0	1	0	0				
P3	4	2	1	5	5	3	2	6	1	0	1	0				
P4	2	1	0	4	3	2	1	1	2	1	1	2				

i)What is the current state of the system?

ii)How many of these request can be granted?

(c) Consider a multithreaded web server running on a machine with N parallel CPU cores. The server has M worker threads. Every incoming request is put in a request queue, and served by one of the free worker threads. The server is fully saturated and has a certain throughput at saturation. Under which circumstances will increasing M lead to an increase in the saturation throughput of the server?(2 marks)

Q 3 (a) Consider a system with only virtual addresses, but no concept of virtual memory or demand paging. Define total memory access time as the time to access code/data from an address in physical memory, including the time to resolve the address (via the TLB or page tables) and the actual physical memory access itself. When a virtual address is resolved by the TLB, experiments on a machine have empirically observed the total memory access time to be (an approximately constant value of) t_h. Similarly, when the virtual address is not in the TLB, thetotal memory access time is observed to be t_m. If the average total memory access time of thesystem (averaged across all memory accesses, including TLB hits as well as misses) is observed to be t_x, calculate what fraction of memory addresses are resolved by the TLB. In other words, derive an expression for the TLB hit rate in terms of t_h, t_m and t_x. You may assume $t_m > t_h$. (6 marks)

(b) Consider a system with several running processes. The system is running a modern OS that uses virtual addresses and demand paging. It has been empirically observed that the memory access times in the system under various conditions are: t1 when the logical memory address is found in TLB cache, t2 when the address is not in TLB but does not cause a page fault, and t3 when the address results in a page fault. This memory access time includes all overheads like page fault servicing and logical-to-physical address translation. It has been observed that, on an average, 10% of the logical address accesses result in a page fault. Further, of the remaining virtual address accesses, two-thirds of them can be translated using the TLB cache, while one-third require walking the page tables. Using the information provided above, calculate the average expected memory access time in the system in terms of t1,t2, and t3.(4 marks)

Q 4 (a) What is the purpose of access matrix and how is it implemented?(5 marks)

(b) Differentiate between Cyclic and acyclic directories on the basis of structure and functionality (5 marks)

Q5 (a) Differentiate between the following (Any two)(3 marks each)

(i) global and local page replacement algorithms

(ii) Virus and worms

(iii) Symmetric and asymmetric encryption

(b) Define an operating system. Enlist the properties of an operating systems and various computing environments in which they are used. (4 marks)

Q6 (a) Is it necessary for threads in a process to have separate stacks? please explain your answer(2 marks)

(b) Is it necessary for threads in a process to have separate copies of the program executable? Please explain your answer (2 marks)

(b) Consider the readers and writers problem. We wish to implement synchronization between readers and writers, while giving preference to writers, where no waiting writer should be kept waiting for longer than necessary. For example, suppose reader process R1 is actively reading. And a writer process W1 and reader process R2 arrive while R1 is reading. While it might be fine to allow R2 in, this could prolong the waiting time of W1 beyond the absolute minimum of waiting until R1 finishes. Therefore, if we want writer preference, R2 should not be allowed before W1. Your goal is to write down pseudocode for read lock, read unlock, write lock, and write unlock functions that the processes should call, in order to realize read/write locks with writer preference. You must use only simple locks/mutexes and conditional variables in your solution. Please pick sensible names for your variables so that your solution is readable
(6 marks)

***** **THE END** *****