

# POORNIMA

## COLLEGE OF ENGINEERING

### Assignment Sheet-2

Course: B Tech.

Name of Faculty: Amol Saxena

Date of preparation: 06-08-2019

Class/ Section – III Yr (V Sem.) - IT

Name of Subject: Operating System (Sub Code:5IT4-03)

Date of submission: 13-08-2019

Q. No.	Question	CO	PO	PSO
1	Consider a logical address space of 64 pages of 1024 words each, mapped onto a physical memory of 32 frames. a. How many bits are there in the logical address? b. How many bits are there in the physical address?	CO1	PO1	PSO1
Sol.	a. $2^6 = 64 + 2^{10} = 1024 = 16 \text{ bits.}$ b. $2^5 = 32 + 2^{10} = 1024 = 15 \text{ bits.}$			
2	Assume that you have a page-reference string for a process with m frames (initially all empty). The page-reference string has length p; n distinct page numbers occur in it. Answer these questions for any page-replacement algorithms: a. What is a lower bound on the number of page faults? b. What is an upper bound on the number of page faults?	CO3	PO2	PSO2
Sol.	a. The lower bound is n, the number of distinct page numbers. Each page will always be faulted for at least once, when it is first brought into memory. b. If $m \geq n$ (the number of frames is greater than number of distinct pages) then the upper bound on the number of page faults is n. We will fault once for each page. If $m < n$ (that is, if the number of frames is less than the number of distinct pages) then the upper bound is p. A really bad page replacement algorithm might fault for every page. For example, consider what would happen if a program was to linearly access a very large array over and over again and the kernel was using a pure LRU page replacement algorithm.			
3	Consider a paging system with the page table stored in memory. a. If a memory reference takes 220 nanoseconds, how long does a paged memory reference take? b. If we add TLBs, and 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes zero time, if the entry is there.)	CO3	PO2	PSO2
Sol.	a. 440 nanoseconds; 220 nanoseconds to access the page table and 220 nanoseconds to access the word in memory. b. Effective access time = $0.75 * (220 \text{ nanoseconds}) + 0.25 * (440 \text{ nanoseconds}) = 165 + 110 = 275 \text{ nanoseconds}$			
4	How does semaphores solve dining philosophers problem?	CO3	PO3	PSO3
Sol.	Use an array of semaphores <i>chopstick</i> [0..4] all initialized to 1.			

Dr. Mahesh Bunde  
B.E., M.E., Ph.D.  
Director  
Poornima College of Engineering  
131-0, FIICO Institutional Area  
Ghatapada, JAIPUR

	<p><b>repeat</b></p> <pre>wait(chopstick[i]); wait(chopstick[(i+1) mod 5]); ... eat ... signal(chopstick[i]); signal(chopstick[(i+1) mod 5]); ... think ...</pre> <p><b>until false;</b></p> <p>Can suffer from <b>deadlock</b> (e.g. all philosophers decide to eat at the same time and all pick up their left chopstick first) and/or <b>starvation</b>. Some ways to avoid deadlock are:</p> <ol style="list-style-type: none"> <li>1. Don't allow all philosophers to sit and eat/think at once.</li> <li>2. Pick up both chopsticks in a critical section</li> <li>3. Alternate choice of first chopstick</li> </ol>															
5	<p>Consider a demand-paging system with a paging disk that has an average access and transfer time of 20 milliseconds. Addresses are translated through a page table in main memory, with an access time of 1 microsecond per memory access. Thus, each memory reference through the page table takes two accesses. To improve this time, we have added an associative memory that reduces access time to one memory reference, if the page-table entry is in the associative memory.</p> <p>Assume that 80 percent of the accesses are in the associative memory, and that, of the remaining, 10 percent (or 2 percent of the total) cause page faults. What is the effective memory access time?</p>	CO3	PO3	PSO3												
Sol.	<p>Access Time = (0.8) * (1 microsecond) + (0.18) * (2 microsecond) + (0.02) * 20002 microsecond)</p> <p>= 401.2 microsecond</p> <p>= 0.4 millisecond</p>															
6	<p>Give a Semaphores solution to the following precedence problem. It is required that the statement A2 in process P1 to complete before statement B4 in process P2 begins (A2&lt;B4).</p> <table> <tr> <td>Process P1</td> <td>Process P2</td> </tr> <tr> <td>A1;</td> <td>B1;</td> </tr> <tr> <td>A2;</td> <td>B2;</td> </tr> <tr> <td>A3;</td> <td>B3;</td> </tr> <tr> <td>A4;</td> <td>B4;</td> </tr> <tr> <td>A5;</td> <td>B5</td> </tr> </table>	Process P1	Process P2	A1;	B1;	A2;	B2;	A3;	B3;	A4;	B4;	A5;	B5	CO4	PO3	PSO1
Process P1	Process P2															
A1;	B1;															
A2;	B2;															
A3;	B3;															
A4;	B4;															
A5;	B5															

Sol.	<b>Semaphore s=0;</b> Process P1 A1; A2; <b>Signal(s);</b> A3; A4; A5;	Process P2 B1; B2; B3; <b>Wait(s)</b> B4; B5			
------	---	--	--	--	--