United International University

Department of Computer Science and Engineering

CSI 309/CSE 4509: Operating System Concepts
Final Examination
Summer 2021

Time: 1 Hour 30 Minutes **Full Marks: 25**

[Any examinee found adopting unfair means will be expelled from the trimester/program as per UIU disciplinary rules.]

[**Answer all the questions. Figures in the right margin indicate marks.]**

1. a. Why do we need page replacement algorithms? [1+3]

   For the given reference string below, apply the Optimal page-replacement algorithm for 5 initially empty page frames. Show the details of what pages are in those 5-page frames and use (x) to mark when a page fault occurs. Find the number of page faults that may occur in this case.

   Reference String: **1 4 3 8 4 6 4 7 3 5 3 1 8 7**

   b. Assume that the system's overall memory access time is 20 nanoseconds. For 10000 instructions, there is page fault overhead on both sides =10000 nanoseconds (approx.). It takes 4000000 nanoseconds to swap pages in. Around 50% of time it needs to swap the page out which takes 2000000 nanoseconds. Now, calculate the **Effective Access Time (EAT)** for a page fault rate of *p*=0.2? [2]

2. a. Explain **Race Condition** with the following statements for process *P1* and *P2*. Here *P* is a shared variable with initial value=3. How many different values of "*P*" you can get? Execute the instructions in order: [2]

   First order: [i, ii, iii, iv]
   Second order: [i, iii, iv, ii]
   Third order: [iii, i, iv, ii]

P1()	*P2()*
{	{
i. Q=P-1;	iii. R=2*P;
ii. P=2*Q;	iv. P=R-1;
}	}

b.

Code snippet for wait()	Code snippet for signal()

wait (S) {
while (S <= 0); // no-operation
S--;
}
	signal (S) {
	S++;
	}

[4]

Prove the **Mutual Exclusion** and **Progress** criteria of synchronizing various processes in critical section using **Binary Semaphore (*S*)** technique for process P_1, P_2…….P_N. *S* is initialized with 1. Prove for the following scenario:

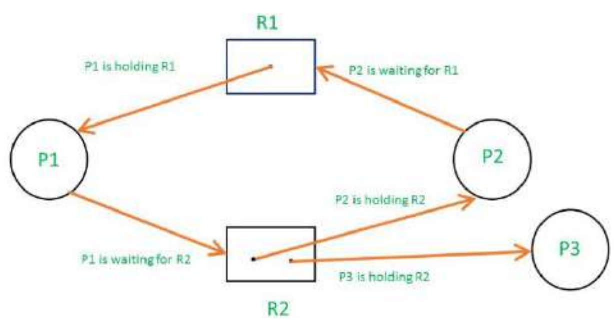
i. P_1 is in critical section but P_2 wants to enter into critical section.
ii. P_1 has finished its task and P_2 again wants to enter into critical section.

3. a. Consider the following condition of an operating system in the following table. [5]

Process	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	1	1	2	4	3	3	2	1	0
P1	2	1	2	3	2	2			
P2	4	0	1	9	0	2			
P3	0	2	0	7	5	3			
P4	1	1	2	1	1	2			

i. Calculate the content of the need matrix?
ii. Determine the total amount of resources of each type?
iii. Is the system in a safe state? (Note: Check for safe sequence)

b. For the following Resource Allocation Graph (RAG), fill up the request and allocation table for process *P1*, *P2* and *P3* provided below, regarding resource type *R1* and *R2*. Is there any deadlock occurring? [2]



Process	Allocation		Request	
	Resource		Resource	
	R1	R2	R1	R2
P1				
P2				
P3				

4. a. Calculate the total number of head movements using Circular LOOK (C-LOOK) and SCAN disk scheduling algorithm for the following requests queue: 82, 95, 178, 36, 112, 15, 123, 53, 64. The current head position of read/write is 50. Draw the chart by moving from left to right. [2+2]

   b. Answer the following regarding file blocks allocation strategies: [1+1]

      i. Which type of fragmentation occurs for contiguous allocation of file blocks.

      ii. Why indexed allocation suffers from pointer overhead issue?