United International University (UIU)

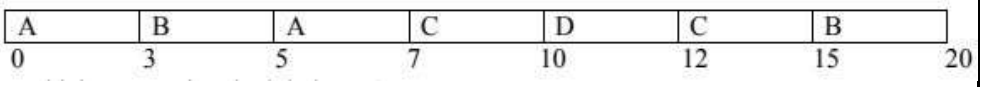
Dept. of Computer Science & Engineering (CSE)

Final-Term Exam. : : Trimester: Fall- 2017

Course Code: CSI 309 **Course Title: Operating System Concepts** **Sec: B**

Total Marks: 40 Duration: 2 hours

There are Five (5) questions. **Answer any four (4) questions**. Figures are in the right-hand margin indicate full marks.

Ques-1		
a.	Suppose the time periods on Shortest Process Next Scheduling is T_0, $T_1, T_2, ...T_n$. The first estimate of runtime is considered to be $E_1 = aT_0+(1-a)T_1$, where a = ⅓. Find out E_3, the third estimate of runtime	2
b.	In a system using Guaranteed Scheduling, the processes A, B, C, D are promised to have the following percentage of the CPU. A:20% B:30% C:40% D:10% The following schedule is observed. Which process should be scheduled next?  	4
c.	Consider the following jobs with arriving times given as 1, 3, 4 , 6, 7, 8, and their duration as 2, 4, 3, 2, 1, 5. Draw a Gantt chart of the round-robin scheduling by considering the time quanta to be 1s.	4
Ques-2		

a.	What is meant by critical region? To protect the critical region, the lock variable is used on the following code:
	`int lock = 0;`
	`while (lock);`
	`lock = 1;`
	`//EnterCriticalSection;`
	`  access shared variable;`
	`//LeaveCriticalSection;`
	`lock = 0;`
	
	Does this code work? Explain.
		1+2

b.	(a) A fast food restaurant has four kinds of employees: (i) order takers, who take customers' orders; (ii) cooks, who prepare the food; (iii) packaging specialists, who stuff the food into bags; and (iv)cashiers, who give the bags to customers and take their money. Each employee can be regarded as a process. Write code to maintain the synchronization among these	3

<table>
<tr><td></td><td>processes.</td><td></td></tr>
<tr><td>c.</td><td>Consider an ATM cubicle with 4 ATM booths. Use semaphore to make sure that at most 4 people can access the 4 booths simultaneously.</td><td>3</td></tr>
<tr><td>d.</td><td>What is race condition?</td><td>1</td></tr>
<tr><td>Ques-3</td><td></td><td></td></tr>
<tr><td>a.</td><td>Suppose, there are five processes and four resources. The processes are P0, P1, P2, P3, P4 and P5, and resources are A, B, C and D. Each resource has multiple instances. At a certain time, operating system has the following scenario.
<table>
<tr><th rowspan="2">Processes</th><th colspan="4">Max</th><th colspan="4">Allocation</th><th colspan="4">Available</th></tr>
<tr><th>A</th><th>B</th><th>C</th><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>P1</td><td>4</td><td>2</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td><td>0</td><td rowspan="5">1</td><td rowspan="5">1</td><td rowspan="5">2</td><td rowspan="5">1</td></tr>
<tr><td>P2</td><td>3</td><td>4</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>0</td></tr>
<tr><td>P3</td><td>1</td><td>4</td><td>3</td><td>1</td><td>1</td><td>2</td><td>2</td><td>0</td></tr>
<tr><td>P4</td><td>3</td><td>3</td><td>0</td><td>2</td><td>1</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>P5</td><td>2</td><td>1</td><td>4</td><td>2</td><td>1</td><td>1</td><td>2</td><td>2</td></tr>
</table>
Now, answer the following questions based on above scenario and Banker’s Algorithm.
i. Is there any safe sequence? If yes, find that; otherwise, explain your answer. Show updated calculated in each step of the Algorithm.
ii. What will happen if process P3 has (5, 6, 5, 5) instances in Max matrix instead of (1, 4, 3, 1)? Explain.
iii. Will you get a safe sequence if Available matrix has (1, 0, 1, 1) instead of (1, 1, 2, 1)?</td><td>5

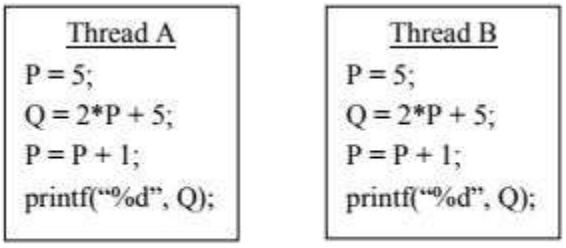
3

2</td></tr>
<tr><td>Ques-4</td><td></td><td></td></tr>
<tr><td>a.</td><td>What is deadlock? Illustrate with an example.</td><td>2</td></tr>
<tr><td>b.</td><td>Write down and briefly explain the four conditions for deadlock.</td><td>2</td></tr>
<tr><td>c.</td><td>Suppose we have a system with 6 processes (A, B, C, D, E, F) and 4 resources (R1, R2, R3, R4). The following statements define the current scenario.
i. Process A is waiting for R1 and R4.
ii. Process B holds R2 and is waiting for R1.
iii. Process C holds R3 and is waiting for R2 and R1.
iv. Process D is waiting for R1 and R4.
v. Process E holds R4 and is waiting for R3.
vi. Process F holds R1 and is waiting for R3.
Draw the Resource Allocation Graph for the current scenario.

Now, execute the following operations and draw Resource Allocation Graph again.
i. Process B releases R2</td><td>3

1</td></tr>
</table>

d.	void put_forks(i)
	{
	down(&mutex);
	state[i] = THINKING;
	test(LEFT);
	test(RIGHT);
	up(&mutex);
	}
	On the able code snippet from the dining philosopher problem, what is the purpose of the function calls: test(LEFT) and test(RIGHT)?
		2

Ques-5		
a.	 N.B.: Consider all assignment operations are atomic and thread can switch context. i. Is there any race condition in the above code? Please explain. ii. Detect the critical section in the above code.	1 1
b.	Draw the file system layout. Define the term inode.	1+1
c.	Suppose, you have a disk of size 10 MB where each block is of size 1 MB. Now you want to create 3 files: A (2.5 MB), then B (3.6 MB), then C (2 MB). Now show the disk block allocations using the following schemes: (i) Contiguous Allocation (ii) Linked List Allocation (iii) Linked list allocation using a table in memory	1 2 1
d.	Suppose, "assignment.txt" is a shared file which is owned by user A and A shares this file with user B via symbolic linking. Now after they have done with the homework, owner A removes the file. Is the file actually deleted from the file system? Will any attempt to read the file by user B succeed?	1+1