



Daffodil International University

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Final Exam Examination, fall 2020 @ DIU Blended Learning Center

Course Code: CSE323 (Eve), Course Title: Operating Systems

Level: 3 Term: 2 Section: O-2 Eve

Instructor: JAP Modality: Open Book Exam

Date: Saturday 24 December, 2020 Time: 06:30-10:30PM

Four hours (4:00) to support online open/case study based assessment Marks: 40.0

Answer all of the questions.

1. a. Draw a resource allocation graph according to the following resource matrix. Determine if there is a deadlock. If yes, indicate the processes and resources involved and add one resource which could fix the deadlock. If not, argue why this is the case, and provide a feasible sequence of processes to show completion. 6.0

	Allocation				Need				Available			
	R0	R1	R2	R3	R0	R1	R2	R3	R0	R1	R2	R3
P1	0	0	0	1	2	0	0	0	2	0	0	1
P2	0	1	1	0	0	1	0	1				
P3	0	1	0	0	0	0	1	1				
P4	1	0	1	0	0	1	0	0				
P5	1	0	0	0	0	0	0	1				

- b. Suppose a system uses Bankers algorithm to avoid deadlock. Consider there are seven processes in the system. They need four types of resources: A, B, C, and D. They have some resources and need more resources to complete their task. The system has 3 instances of A, 2 instances of B, 0 instances of C and 4 instances of D free resources. In the Allocation and Max matrix the value resources of the process will be the last four digits of your own DIU ID. 6.0
- For example**, if a student named John has ID 191-15-45678, then the value for A is 5, B is 6, C is 7 and D is 8 for process P2.

Processes	Allocation				Maximum			
	A	B	C	D	A	B	C	D
P1	2	2	2	0	5	4	5	2
P2								
P3	3	4	0	1	6	5	0	4
P4	0	1	2	2	5	7	4	3
P5	3	4	1	1	4	5	4	5
P6	4	5	2	3	6	7	2	4
P7	3	2	3	3	3	4	5	3

Now see the following snapshot and give the answer of the following questions.

- I. Find out the content of the Need of each process.
- II. Find out the total value of instances of different resources at the initial stages.
- III. Suppose two requests from two processes come. One request from process P1 for (2, 2, 0 and 1) resources and another request from P4 for (5, 3, 3 and 1) resources. Can these requests be granted immediately ensuring the system is safe? If yes, show the safe sequence. If not, indicate the processes and resources involved and add required resources which could fix the problem.

2 a. Suppose 1MB memory is partitioned into some fixed sized blocks each block having a size of 128KB. The list of request orders is: 50k, 150k, 90k, 130k, 70k, 80k, 120k, 180k, and 60k. Do you think it is possible for an operating system to serve all the requests by implementing the memory allocation algorithms? Justify your answer. 5.0

b. Now suppose you have given the same request order with 4 variable sized memory partitions: 100k, 200k, 80k, and 110k. Also consider the number of turns for each process is between 1 and 3. Which memory allocation algorithm will provide the best solution in this case? Explain with proof. 8.0

3. a. Consider the following reference string and frame:
1 2 3 1 5 2 7 3 8 9 1 8 2 1 7 3 5 8 5 1 2 8 9 6.0

1
7

	<p>i. Do you think LRU is better than an Optimal algorithm while finding page faults in operating systems? Explain your answer with proper justification. Now implement the better algorithm for the above scenario and find out page faults and page hits.</p>	
b.	<p>Suppose a disk drive has 2500 tracks (0-2499) and the disk queue having I/O requests in the following order as follows:</p> <p>90, 890, 2400, 700, 2100, 1700, 800, 300, 150, 200, 50, 550, 250 and 2000</p> <p>Current position of the Read Write head is 500 and assume that previous request was the first three digits of your id [Sample: Id 192-15-34567 ; previous request 192].</p> <p>For minimizing seek distance which disk scheduling algorithm will you use? Why? Explain your answer. Now calculate the seek distance for your chosen algorithm.</p>	4.0
c.	<p>What have you learned from this course? Is this course helpful to you? Give a brief description of it</p>	5.0

Best of Luck ☺