Date: May 6, 2016 **Marks:** 65 **Time:** 150 min

Instructions:

1. Submit the objective part before starting this exam.
2. Try to solve question 1 on page 1, question 2 on page 2, and so on. Only the first 10 pages of the answer sheet will be marked.

Question 1 (10 marks)

As you know, the files are stored in the form of blocks. Every block of a file has a logical block number. The first block has the logical block number 0, the second has no 1, the third equal to 2, and so on and so forth. Assume a file named "outline.pdf" is stored in a FAT based filesystem, andits first block is stored at physical block no5.(Means its 0th logical block number maps to physical block number 5).The rest of the file-blocks can be reached by looking at the following file allocation table:

6	EOF	7	8	9	2	4	0	1	3
0	1	2	3	4	5	6	7	8	9

Now, translate the following logical block numbers into the corresponding physical block numbers:

(a) 9 (b) 6 (c) 2

Question 2 (5 + 5 marks)

(a) Show execution of the Optimal page replacement algorithm on the following page-reference string, assuming four frames:

1 2 3 4 1 5 6 7 8 1 2 3 4

(b) Repeat part (a) using the LRU algorithm.

Question 3 (5 + 5 marks)

Question 2 (5 + 5 marks)

(a) Show execution of the Optimal page replacement algorithm on the following page-reference string, assuming four frames:

1 2 3 4 1 5 6 7 8 1 2 3 4

(b) Repeat part (a) using the LRU algorithm.

Question 3 (5 + 5 marks)

Consider the following page table of a process P:

Frame#	50	20	70	10	90	30	80	40	60	55
Page#	0	1	2	3	4	5	6	7	8	9

Now, convert the following logical addresses into the corresponding physical addresses, assuming a page size of 1000 bytes (not 1024):

(a) 3500 (b) 7400

Operating Systems
Final Exam, Spring 2016

Date: May 6, 2016 **Marks:** 65 **Time:** 150 min

Question 4 (5 marks)
Consider the following formulas to compute process priority in a system:

Formula A: priority = (CPU used in last two seconds)
Formula B: priority = 1 / (CPU used in last two seconds)

Assume there is an inverse relationship between priority and its value; for example 10 is higher than 20 ... Which of the two formulas is better? Give reason for your answer.

Question 5 (10 marks)

In the following applications/programs, where shall we use threads, and where shall we use processes? Give reasons for your answers. Be precise and concise.

a) Searching or sorting a large array
b) A program that executes different applications during its execution
c) An internet browser that can load multiple web pages in its different tabs

Question 6 (5+5 marks)
(a) The following three threads (note that there are exactly 3 threads, no more no less) generate a string containing characters a, b and c in an arbitrary order. Following are few examples:
[illegible] abababcbcbcbaaabbbc, cccccbabababababababacccacb

processes? Give reasons for your

a) Searching or sorting a large array
b) A program that executes different applications during its execution
c) An internet browser that can load multiple web pages in its different tabs

Question 6 (5+5 marks)

(a) The following three threads (note that there are exactly 3 threads, no more no less) generate a string containing characters a, b and c in an arbitrary order. Following are few examples:

abcbaccccbaab, abababcbcbcbaaabbbc, ccccbababababababababacccacb

// thread 1
while (1){
cout<< "a";
}
	// thread 2
	while (1){
	cout<< "b";
	}
		// thread 3
		while (1){
		cout<< "c";
		}

Synchronize the threads using semaphores so that the printed string is a concatenation of the substring "cba". Here are few examples:

cba, cbacba, cbacbacba, ...

(b) Now again synchronize the threads in part (a) such that instead of infinite concatenation of substring "cba" an infinite concatenation of substring "cbbba" is generated ... Note that you are not allowed to add or modify any **cout** statement.

Operating Systems
Final Exam, Spring 2016

Date: May 6, 2016 **Marks:** 65 **Time:** 150 min

Question 7 (10 marks) **(For Section C and D Only)**

Write a C/C++ program that executes any given two programs using the Unix (Linux) Pipeline. The program takes names of any two existing programs as input. It then executes the first program, while redirecting its output to a pipe. Finally, it executes the second program with its input redirected to the same pipe.

Note that your design shall support as much concurrency (parallelism) as possible. The second program shall start as soon as possible; it shall not wait for termination of the first program.

Question 7 (10 marks) **(For Section A and B Only)**

(a) Following table shows the arrival time of four processes.

Process Name	Arrival Time
P1	1
P2	5
P3	7
P4	9

We assume that there are 3 resources R1, R2and R3, which are free initially. Following tab

as possible; it shall not wait for termination of the first program.

Question 7 (10 marks) **(For Section A and B Only)**

(a) Following table shows the arrival time of four processes.

Process Name	Arrival Time
P1	1
P2	5
P3	7
P4	9

We assume that there are 3 resources **R1**, **R2**and **R3**, which are free initially. Following table shows the requests from processes, fill on the right side of following table, the results of these requests using wait and die algorithm.

Request	State of Resources
	R1, R2, R3 are free
P1 requests R1	R1 is assigned to P1
P2 requests R2	
P1 releases R1	
P3 requests R1	
P4 requests R3	
P4 requests R1	
P1 requests R3	