جامعة الملك سعود
King Saud University
College of Computer & Information Sciences
Computer Science Department

Course Code: CSC 227	**Final Exam**
Course Title: Operating Systems	May 7, 2020 Ramadan 14, 1441
Semester: Second Semester 1441-1442	09:00- 11:59 AM
Answer Sheet	
Student Name:	
Student ID:	
Section No.	
Student Serial No:	**20**

Course Learning Outcomes	**Question No.**	**Points**	**Student's Points**
CLO5 [Process Synchronization]	Q1	6	
CLO6 [Memory Management]	Q2	3	
CLO6 [Memory Management]	Q3	2	
CLO6 [Memory Management]	Q4	2	
CLO6 [Virtual Memory]	Q5	3	
CLO6 [Virtual Memory]	Q6	2	
CLO6 [Virtual Memory]	Q7	2	

Question 1 [6 Marks, 0.5 each]**:** Choose ONLY ONE correct answer for each of the following problems, then fill your answer in the box below.

Problem Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Student Answer												

Consider the following attempt to solve the critical section problem for problems 1, 2, and 3:

boolean lock = false; //global variable

P0	**P1**

while (true){
    while (lock) ; //(1)
    lock = true; //(2)
    // critical section (3)
   lock = false; //(4)
  //reminder section (5)
    }
	while (true){
	    while (lock) ; //(1)
	    lock = true; //(2)
	    // critical section (3)
	   lock = false; //(4)
	  //reminder section (5)
	    }

1. Select the property that best describes the attempt:
   a. The attempt solves the critical section problem correctly.
   b. The attempt violates the mutual exclusion property.
   c. The attempt violates the progress/bounded wait property.
   d. All of the above.

2. Which line number(s) specifies the Entry Section?
   a. Line (1)
   b. Line (2)
   c. Line (4)
   d. Lines (1) and (2)

3. Which line number(s) specifies the Exit Section?
   a. Line (1)
   b. Line (2)
   c. Line (4)
   d. Lines (1) and (2)

4. Consider the following attempt to solve the critical section problem:

boolean intendToEnter[] = {false, false}; //global variable

P0	P1

while (true) {
 while (intendToEnter[1]) ;
 intendToEnter[0] = true;
 	
//critical section
 	
 intendToEnter[0] = false;
 //reminder section
}
	while (true) {
	 while (intendToEnter[0]) ;
	 intendToEnter[1] = true;
	
	 // critical section
	
	 intendToEnter[1] = false;
	 // reminder section
	 }

Select the property that best describes the attempt:

a. The attempt solves the critical section problem correctly.
b. The attempt violates the mutual exclusion property.
c. The attempt violates the progress/bounded wait property.
d. All of the above.

5. Consider the following attempt to solve the critical section problem:

boolean intendToEnter[] = {false, false}; //global variable

P0	P1

while (true) {
 intendToEnter[0] = true;
 while (intendToEnter[1]) ;
 	
 //critical section
 	
 intendToEnter[0] = false;
 //reminder section
}
	while (true) {
	 intendToEnter[1] = true;
	 while (intendToEnter[0]) ;
	
	 // critical section
	
	 intendToEnter[1] = false;
	 // reminder section
	 }

Select the property that best describes the attempt:

a. The attempt solves the critical section problem correctly.
b. The attempt violates the mutual exclusion property.
c. The attempt violates the progress/bounded wait property.
d. All of the above.

6. Consider the following cooperating processes deposit and withdraw for a banking application:

int balance = 0; //global variable

deposit (){	withdraw (){
int new_balance;	int new_balance;
new_balance = balance;	new_balance = balance;
new_balance++;	new_balance--;
balance = new_balance;	balance = new_balance;
}	}

The processes corresponding to deposit() and withdraw() executed for one time each concurrently. What is the value of the shared variable balance after completion of these two processes?

a. balance = 0 or balance = 1.
b. balance = 1 or balance = 2.
c. balance can take any integer value in a range between -1 and 1 (inclusive).
d. None of the above.

7. Assume that sum is an integer variable. Which of the following operations are atomic?

a. sum ++;
b. sum - -;
c. Both (a) and (b).
d. None of the above.

8. Which of the following is NOT true for Peterson's solution?

a. Mutual exclusion is preserved.
b. The progress requirement is satisfied.
c. The bounded-waiting requirement is met.
d. Peterson's solution works for synchronization among an arbitrary number of processes.

9. An airline reservation system with two kinds of processes; the first kind search for available seats, and the second type makes reservations for travelers. This system is considered a practical example of the following problem.

a. Producer-consumer problem.
b. Readers-writer problem.
c. Dining philosopher's problem.
d. None of the above.

Consider the following scenario and code to solve the critical section problem for problems 10, 11, and 12:

Al Qiddiya is a major entertainment project in Riyadh. It includes resorts, parks and a city center. Inside its parks, visitors are required to navigate using golf carts. Each golf cart can carry only one visitor. Visitors walk around the park for a while, then line up to take a golf cart ride. If the cart is available, the visitor rides and drives through the park for a random amount of time. If none of the n carts are available to ride (all occupied by other visitors), then a visitor who wants to ride waits. Similarly, if a cart is available but there are no ready visitors to ride, then the cart waits. Assume that you have m visitor processes and n cart processes. Initially when the park opens, there are n carts and no visitors yet. Further, assume that two semaphores are used to synchronize these processes with busy waiting: visitor and cart in the given algorithms below.

Semaphore visitor = __, cart = __;

visitor_p()	cart_p()

// walk around park
// I want to ride
 	
__________; ---(LINE-A)
wait (cart); ---(LINE-B)
 	
// ride the cart
	// idle until someone arrives
	
	wait (visitor);---(LINE-C)
	
	// carry the visitor for a random time
	
	//done riding
	__________; ---(LINE-D)

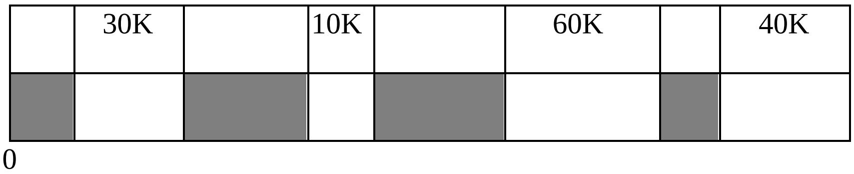
10. What should be the initial value of the visitor and cart semaphores?
    a. visitor = 0, cart = 0.
    b. visitor = m, cart = n.
    c. visitor = n, cart = m.
    d. visitor = 0, cart = n.

11. What is the proper synchronization code in LINE A?
    a. signal(cart).
    b. signal(visitor).
    c. wait(cart).
    d. wait(visitor).

12. What is the proper synchronization code in LINE D?
    a. signal(cart).
    b. signal(visitor).
    c. wait(cart).
    d. wait(visitor).

Question 2 [3 Marks]**:**

PART A [1 Mark, 0.5 each]**:** Consider the following memory image:

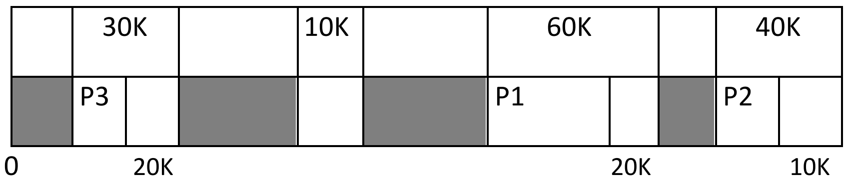
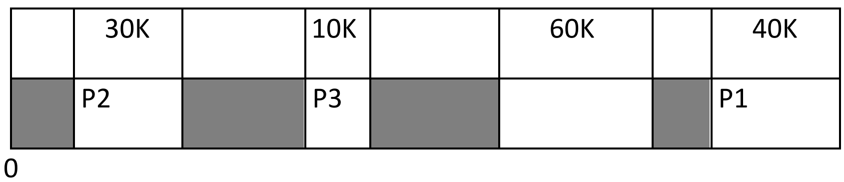


Given that:

- Blank areas represent non-occupied areas in the memory.
- Shaded areas represent occupied areas in the memory.
- The memory address (0) represents the start of allocation of the process in the memory.
- The search starts always from the start of the memory.
- Memory allocation is based on **variable-sized partition**.

If processes arrive in order P1, P2, and P3 of sizes 40K, 30K, and 10K respectively, and are allocated memory as shown below, indicate which allocation algorithm each of the following solutions depicts. Choose ONLY ONE correct answer for each of the following solutions, then fill your answer in the table below.

a) First-fit **b)** Best-fit **c)** Worst-fit

Solution	**Allocation algorithm**
	c
	b

PART B [2 Marks, 1 each]**:** Consider the following page table:

Page #	**Frame #**
0	5
1	2
2	6
3	3

Given that:

- The frame size is 1024 bytes.
- The logical addresses are in the format **<x, y>** where *x* represents that page number and *y* represents the offset number (0-1023 different offsets).

Translate -if possible- the following logical addresses into their physical addresses. Indicate if an illegal addressing occurs (*Note: You are required to show the detailed steps of your solution*):

Logical Address	**Physical Address or Illegal address**
a) <2, 80>	<6, 80>
b) <3,1200>	Illegal address

Question 3 [2 Marks, 0.5 each]**:**

Consider a logical address space of 64 pages with a 2KB page size, mapped onto a physical memory of 16 frames. Answer the following questions:

a) How many bits are required for the page number?

Solution	
Detailed Steps	$2^6 = 64$ # pages $\Rightarrow$ 6
Final Answer	6

b) How many bits are required for the offset?

Solution	
Detailed Steps	$2KB = 2 \times 2^{20} = 2^{21}$ $2^{21} \rightarrow 21$
Final Answer	21

c) How many bits are required in the logical address?

Solution	
Detailed Steps	# Pages $\Rightarrow 2^6$ and offset $\Rightarrow 2^{21}$ $2^6 \times 2^{21} = 2^{27}$
Final Answer	27

d) What is the size of the physical address space?

Solution	
Detailed Steps	# frames $= 16 \Rightarrow 2^4$ and offset $\Rightarrow 2^{21}$ $2^4 \times 2^{21} = 2^{25}$
Final Answer	2^{25}

Question 4 [2 Marks]**:** Find the total context switch swapping time (in millisecond), if a disk drive with a transfer rate of 4 MB/s is used to swap a process that is 512 KB in size?

Solution	
Detailed Steps	swap out $512 / 4 \times 2^{10} = 0.125$ s $\Rightarrow$ 125 ms swap in $2 \times$ = 250 ms
Final Answer	250 ms

~~**Question 5**~~ [3 Marks]**:** Consider the following page reference string:

0 1 2 0 6 0 2 3 4 2 5 6 4

Using the *Least Recently Used (LRU) replacement algorithm*, allocate the process's pages into the below frames. Assuming three frames were allocated for the process?

0	**1**	**2**	**0**	**6**	**0**	**2**	**3**	**4**	**2**	**5**	**6**	**4**

Question 6 [2 Marks]**:** Choose only ONE answer.

1. In a demand paging system, an average page fault service time is 4ms, with page fault rate (p) equal to 0.0003 and a memory-access time of 100 ns. The effective access time (EAT) is:
   a) 4.03 ns
   b) 99.97 ns
   c) 1299.97 ns
   d) None of above

2. Which of the following decides whether or not a system needs to modify a page into the disk when it is replaced?
   a) Base register
   b) Page table
   c) Limit register
   d) Dirty bit

3. Consider a demand paging system in which main memory consists of 4 frames. Suppose pages 0, 1, 2 and 3 are in memory at a given time. The number of times that each of these pages has been referenced is given below:

Page number	Number of times referenced
0	50
1	53
2	45
3	29

If a page needs to be replaced at this time, which page will be selected using ***the most frequently used (MFU)*** page replacement algorithm?

a) Page 0
b) Page 1
c) Page 2
d) Page 3

4. It means that when a process is swapped in, only the pages that it needs are loaded into the memory.

a) Demand segmentation
b) Belady's Anomaly
c) Demand paging
d) Simple paging

Question 7 [2 Marks]**:** Consider a system with a 2KB frame size. Suppose a process P1 of size 128 KB, a process P2 of size 200 KB, and process P3 of size 72 KB are the only three processes running on this system. Let the system have 150 free frames. How many frames should be allocated to P1, P2, and P3 using the *proportional allocation algorithm*?

(Note: show me your calculations step by step)

	Solution
Detailed Steps	P1: 128/2 = 64 frames P2: 200/2 = 100 // P3: 72/2 = 36 //
Final Answer	

END OF EXAM QUESTION