King Saud University

College of Computer and Information Sciences
Computer Science Department

Course Code:	CSC 227
Course Title:	Operating Systems
Semester:	Spring 2017-2018
Type of Examination:	Midterm 2 Exam.

Student Name:	
Student ID:	
Student Section No.	
Instructor Name:	

Instructions:

- This exam has 8 pages.
- Do not use pencil.
- Write clearly and neatly

Tick the Relevant	Computer Science B.Sc. Program ABET Student Outcomes	Question No. Relevant Is Hyperlinked	Covering %
X	a) Apply knowledge of computing and mathematics appropriate to the computer science;		
X	b) Analyze a problem, and identify and define the computing requirements appropriate to its solution	Q.1-Q.2-Q.4	75%
X	c) Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;	Q.3	25%
X	d) Function effectively on teams to accomplish a common goal;		

	Full Mark	Student's Mark
Question No.1	8.0	
Question No.2	4.0	
Question No.3	4.0	
Question No.4	4.0	
Total	20	

King Saud University
College of Computer and Information Sciences
CSC 227: Operating Systems

Total Marks: 25 **Time: 7:00pm – 8:30pm (90 minutes)**
Fall 2015-16 **Name: ………………………………………….………………………**
Midterm Exam I **ID#: ………………………………………………………………………**
Date: 20-Oct-2015 **Section#:…………………… or Teacher Name: ………………………**

- .

Question 1. [6 marks] Select ONLY ONE ANSWER (the best answer). [CLO 4]

Copy your answer for question 1-1 to 1-12 in the table on page2. ONLY THAT TABLE WILL BE GRADED.

1.	**The section that contains the global variables of a process is:**		2.	**The scheduler that selects which processes should be brought into the ready queue is:**
a.	The text section.		a.	**The long-term scheduler.**
b.	The stack section.		b.	The short-term scheduler.
c.	The heap section.		c.	The medium-term scheduler.
d.	**The data section.**		d.	The CPU scheduler.

3.	**Which of the following is NOT a reason for processes cooperation?**		4.	**A process that has terminated, but whose parent has not yet called wait(), is known as:**
a.	**Independency.**		a.	Orphan process.
b.	Information sharing.		b.	**Zombie process.**
c.	Modularity.		c.	Terminated process.
d.	Computation Speedup.		d.	Child process

5.	**In cascading termination:**		6.	**Which of the following queues contains a set of all processes in the system?**
a.	Only children who has exceeded allocated resources are terminated.		a.	**Job queue.**
b.	Only children whose assigned task is no longer needed are terminated.		b.	Ready queue.
c.	**All children are terminated.**		c.	Device queue.
d.	Only the last created child is terminated.		d.	Waiting queue.

7.	**An instruction executed atomically means:**		8.	**One common problem among the methods suggested for critical section problem is:**
a.	**It cannot be interrupted during its execution**		a.	Continuous waiting
b.	It can interrupt any other instruction		b.	**Busy waiting**
c.	It can be executed in kernel mode		c.	Too difficult to program
d.	It can enter critical section anytime.		d.	Take long time to execute

9.	**A non-preemptive kernel is:**
a.	Free of critical section
b.	Free of deadlock
c.	**Free of race conditions in kernel mode**
d.	Free of starvation

10.	**A preemptive kernel:**
a.	Does not allow preemption of process when running in kernel mode
b.	Allows multiprocessing
c.	Does not allow multiprocessing
d.	**Allows preemption of process when running in kernel mode**

11.	**One solution to critical section problem in uniprocessor machine is:**
a.	**Disabling interrupts**
b.	Enabling interrupts
c.	Using machine language instructions instead of semaphores
d.	Using `set_and_test` instruction

12.	**A thread library provides the programmer**
a.	**With an API for creating and managing threads**
b.	With java program for mathematical calculations
c.	With C program for mathematical calculations
d.	With program to create user level passwords

13.	**Which are possible techniques of handling signals in the context of a multi-threaded process**
a.	The system delivers the signal to the thread to which the signal applies
b.	The system, deliver the signal to every thread in the process
c.	A specific thread is assigned to receive all signals for the process
d.	**All the above**

14.	**In a multithreaded process, which of these models will result in the minimum use of memory ?**
a.	Two-level model
b.	One-to-one model
c.	**Many-to-one model**
d.	Many-to-many model

15.	**Asynchronous thread cancellation is,**
a.	The activity of synchronization between two running threads that share data.
b.	When one thread runs and the other remains in hot standby mode, in case it is needed.
c.	**Thread is allowed to terminate another thread immediately without waiting.**
d.	A thread remains in waiting mode until the other thread terminates, then it receives it results and continues execution.

16.	**In deferred thread cancellation, when a thread receives a cancellation request,**
a.	It is allowed to wait until the current I/O is finished
b.	The thread is allowed to terminate after it closes all the opened files
c.	The thread is allowed to terminate itself immediately and close everything at once
d.	**Thread is allowed to continue running until it reaches a cancellation point and terminate itself in an orderly fashion**

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

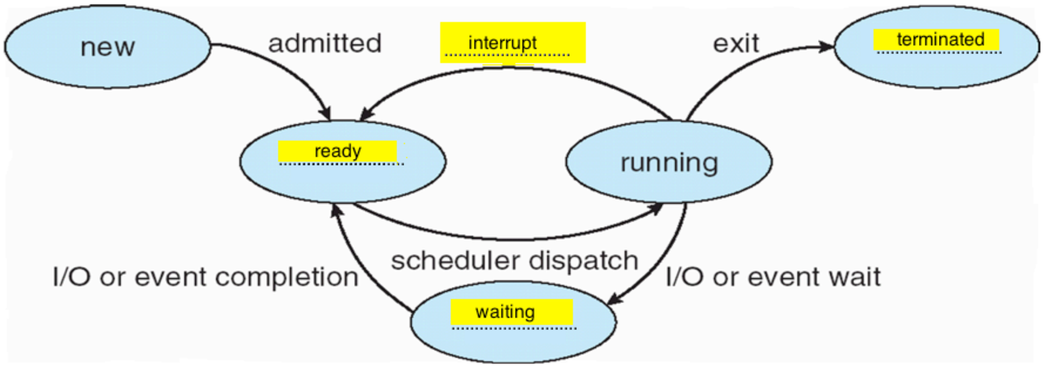
9.	10.	11.	12.	13.	14.	15.	16.

Question 2. [4 marks] [CLO 5]

2-a) [1 marks] Fill in the FOUR blanks in the following process state diagram.

.

Answer:



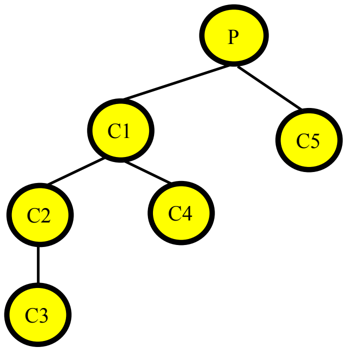
2-b) [3 marks] Consider the following C program: [CLO ??]

```
#include <stdio.h>
int num;
int main(){
      num = 10;
      if (fork() > 0) {
            num = num * 2;
            wait(NULL);
            printf ("num= %d\n", num);
      } else {
            num = num + 5;
            printf("num= %d\n", num);
            fork();
            num++;
            printf("num= %d\n", num);
      }
      fork();
}
```

I. [1 mark] How many processes will be created by this program including the parent process?

6 processes

II. [1 mark] Draw the process tree resulting from this program:



III. [1 mark] What is the output of this program?

```
num= 15
num= 16
num= 16
num= 20
```

Question 3. [4 marks] [CLO 7

3-a) [0.5 marks] What is the critical section problem?

Critical section problem is to design protocol to solve concurrent access to shared data among various processes.

3-b) [1.5 marks] List the requirements that a solution to the critical section problem must satisfy.

1) **Mutual exclusion.**
2) **Progress.**
3) **Bounded waiting.**

3-c) [2 marks] What is starvation? What causes it??

A process may never be removed from the semaphore queue in which it is suspended.

Causes: Deadlock, or LIFO semaphore queue where the rate of wait is higher than signal.

Question 4. [4 marks] [CLO 4]

4-a) [2 marks] Let us consider a process composed of the following CPU bursts and IO bursts:

N∘	**Activity**	**Duration (milliseconds)**		
		CPU	Waiting I/O	Total
1	Starting and initialization	5	10	15
2	Reading Data from a Network	15	20	35
3	Processing Data	50	0	50
4	Saving results to a file	25	15	40
5	Terminating	10	0	10
	Total execution time	105	45	150

1- [0.5 mark] If you want to speed-up this process using multi-threading, which task will you select to implement multi-threading? Why? ……………………….

The task 3, because it is the largest task in terms of CPU activity, and is the one that can give the best time saving.

2- [1 mark] Considering that the system has 2 cores and that only one task can be split into 2 threads, what is the best speedup we can obtain for this process? ……………………….

S = 100/150 = 2/3, P = 50/150 = 1/3

$$= \frac{1}{{}^{2}/_{3} + \frac{{}^{1}/_{3}}{2}} = 1.2$$

3- [0.5 mark] After implementing multi-threading, what will be the best total execution time of this process, from start to the termination. ……………………….

= 150 / 1.2 = 125 ms

4-b) [1 mark] The threads of a same process share part of the process resources. Which resources are different for each thread:

- **Registers**
- **Stack**

4-c) [1 mark] The multi-threading model one-to-one has the advantage of offering a high independence to user threads. What are its disadvantages?

Creating a kernel thread of each user thread takes time
Creating a kernel thread of each user thread uses a lot of memory