Answer the following questions:-

Question No. (1)

A. Multiple Choice: [10 points]

1) In the Round Robin (RR) scheduling; if the time quantum (q) is too large, then:

A) Scheduling is same as First Come First Served (FCFS).
B) More context switches will occur.
C) The average turnaround time decreases.
D) No effect.

2) Any solution to the critical section problem must satisfy the following three criteria:

A) Disabling interrupts, TestAndSet instruction, and semaphores.
B) Request, use and release.
C) Mutual exclusion, no preemption, and circular-wait.
D) Mutual exclusion, progress, and bounded waiting.

3) In First-Come-First-Served CPU scheduling algorithm, when the short processes wait for long process to finish its execution is called:

A) Thrashing.
B) Starvation.
C) Page fault.
D) Convoy effect.

4) Increasing the multiprogramming level can be accomplished by:

A) Packing more processes into memory.
B) Reducing memory fragmentation.
C) Sharing code and data among different processes.
D) All of the above

5) One class of services provided by an operating system is to provide new functionality that is not supported directly by the underlying hardware.

A) **True** B) False

6) If the web server runs as a single-threaded process, it would be able to service many clients at a time.

A) True B) **False**

7) The most important CPU scheduling criteria to be optimized in an interactive operating system is:

A) **Response time.** B) Waiting time. C) Throughput. D) Turnaround time.

8) If all processes are CPU bound, the I/O waiting queues will almost always be empty.

A) **True** B) False

9) When both send( ) and receive( ) are blocking, we have a rendezvous between the sender and the receiver.

A) **True** B) False

10) In the indirect communication, a communication link has the following property:

A) A link is associated with exactly two processes.

B) A link is established only if the pair processes have a shared mailbox.

C) A link is established automatically between every pair of processes.

D) Between each pair of processes, there exists exactly one link.

B. [6 points] Consider the following set of processes to solve the questions I, II and III. (Select the correct answer)

Process	Arrival Time	Burst Time	Priority
F	0	6	5
G	2	3	3
H	3	2	4
I	7	5	2

I) The Gantt chart for Shortest-Job-First (SJF) scheduling algorithm is:

A)

F	F	G	H	H	G	G	F	F	F	F	I	I	I	I	I

B)

F	F	G	G	G	H	H	F	F	F	F	I	I	I	I	I

C)

F	**F**	**F**	**F**	**F**	**F**	**H**	**H**	**G**	**G**	**G**	**I**	**I**	**I**	**I**	**I**

E) None of them

II) The Gantt chart for non-preemptive priority scheduling algorithm is:

(Low number → high priority)

A)

F	**F**	**F**	**F**	**F**	**F**	**G**	**G**	**G**	**I**	**I**	**I**	**I**	**I**	**H**	**H**

B)

F	F	G	G	G	H	H	I	I	I	I	I	F	F	F	F

C)

F	F	F	F	F	F	G	G	G	H	H	I	I	I	I	I

E) None of them

III) The average waiting time using SJF in question I is:

A) 2.5 B) 4 C) 1.75 **D) 3.25** E)None of them

Question No. (2)

1-[3 points] The following events may occur to a process. Identify the starting state it is in at the time of the event and the ending state it transitions to.

Event	Starting state	Ending state
Scheduler dispatches process		
I/O complete		
Process admitted to ready queue for first time		
Scheduler preempts (interrupts) process		
Process finishes execution (task is complete)		
Process initiates I/O		

2-[3 points] Draw a diagram to illustrate the components of the Linux system.

3-[4 points] What is the purpose of system calls? Why we use APIs rather than system calls?

4-[3 points] Clarify what is dual mode operation.

5- List and describe briefly each of the following

A. Three different types of processes scheduler **[3 points]**
B. Four Operating system structures **[4 points]**
C. Four Threading issues **[4 points]**

6- [4 points] Explain what semaphore is and Illustrate using three examples the various types of errors that can be generated when programmers use semaphores incorrectly to solve the critical-section problem.

7- [5 points] Distinguish between message passing model and shared memory model (Make sure to clarify the role of the operating system in each).

8- [5 points] What is the difference between a process and a thread? Describe three different multithreading models for mapping user threads to kernel threads.

9- [6 points] Explain how operating systems have evolved over the years and provide three reasons for this evolution.

10- [5 points] The following pair of processes share a common variable X:

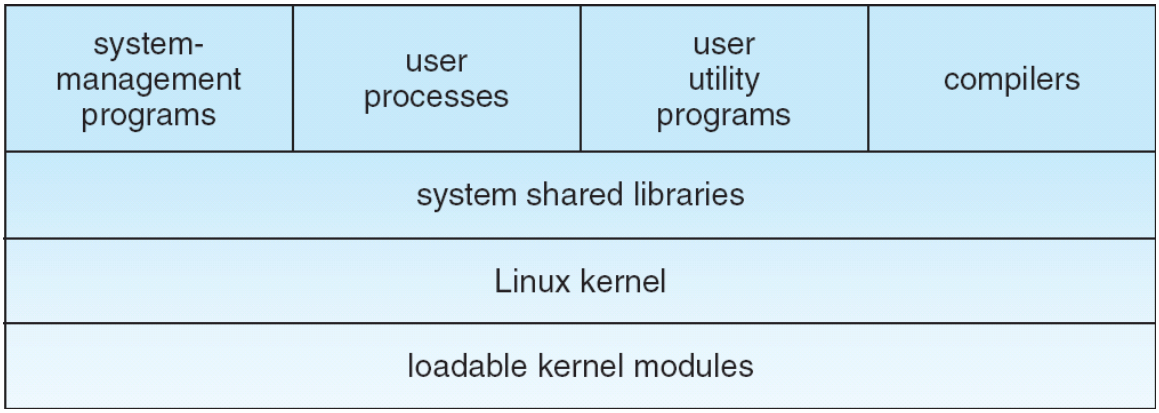
```
Process A                    Process B
    int Y;                       int Z;
A1: Y = X*2;                 B1:  Z = X+1;
A2: X = Y;                   B2:  X = Z;
```

Before either process begins execution X = 5. As usual, statements within a process are executed sequentially, but statements in process A may execute in any order with respect to statements in process B.

A) Describe how a race condition is possible and show how many different values of X are possible after both processes finish executing?

B) Using two semaphores S and T, modify the processes so that the only possible value of X is 6.

With my best wishes



Event	Starting state	Ending state
Program started by user	**Does not exist**	**New**
Scheduler dispatches process	**Ready**	**Running**
I/O complete	**Waiting**	**Ready**
Process admitted to ready queue for first time	**New**	**Ready**
Scheduler preempts (interrupts) process	**Running**	**Ready**
Process finishes execution (task is complete)	**Running**	**Terminated**
Process initiates I/O	**Running**	**Waiting**

- *Semaphore is …*
  - *Semaphore S – integer variable*
  - *That is accessed only through two standard **atomic operations** to modify S: wait() and signal()*
  - *Originally called*
    - *P () for wait() – an atomic operation that waits for semaphore to become positive, then decrements it by one.*
    - *V() for signal()- an atomic operation that increments the semaphore by one , waking up a waiting P, if any.*

- *Examples of misuse:*
  - *mutex is a semaphore which is initialized to 1*
  - *signal (mutex) …. wait (mutex)*
    - *results in a violation of the mutual exclusion requirement*
  - *wait (mutex) … wait (mutex)*

- ◆ *results in deadlock*

- *Omitting of wait (mutex) or signal (mutex) (or both)*
  - ◆ *Either mutual exclusion is violated or a deadlock will occur*

a) What is race condition ? How many different values of X are possible after both processes finish executing?

Race Condition

- Race Condition (a race between two processes who will finish first)
  - Multiple processes or threads read and write data items
  - They do so in away where final result depends on the order of execution of the processes.
  - The output depends on who finishes the race last
- Why do race conditions occur
  - "whenever the state of a shared resource depends on the precise execution order of the processes"
  - Scheduling: Context switches at arbitrary times during execution

A1 A2 B1 B2: X = 11
A1 B1 A2 B2: X = 6
A1 B1 B2 A2: X = 10
B1 A1 B2 A2: X = 10
B1 A1 A2 B2: X = 6
B1 B2 A1 A2: X = 12

b) Using two semaphores S and T, modify the processes so that the only possible value of X is 6.
To ensure a final value of 6, there are two precedence constraints to be enforced: B1 > A2 and A2 > B2. A1 and B1 can happen in any order.

// indicate semaphore initial values here…

Semaphore S = 0 , T = 0 ;

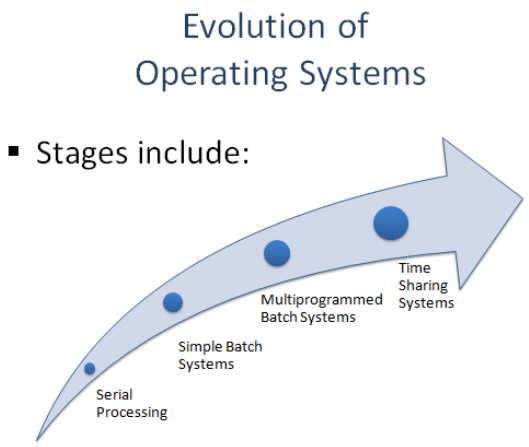
```
Process A
    int Y;
A1: Y = X*2;
wait(S);
A2: X = Y;
signal(T);
```

```
Process B
    int Z;
B1: Z = X+1;
signal(S); wait(T);
B2: X = Z;
```

Threading Issues

- *Semantics of **fork()** and **exec()** system calls*
- *Thread cancellation*
- *Signal handling*
- *Thread local storage*
- *Scheduler activations*

- *A major OS will evolve over time for a number of reasons:*
  - ***Hardware upgrades plus new types of hardware***
  - ***New services: In response to user demand or in response to the needs of system managers, the OS expands to offer new services.***
  - ***Fixes: Any OS has faults. These are discovered over the course of time and*** *fixes are made. Of course, the fix may introduce new faults.*



1. The shared-memory method requires communicating processes to share some variables. The processes are expected to exchange information through the use of these shared variables. In a shared-memory system, the responsibility for providing communication rests with the application programmers; the operating system needs to provide only the shared memory.

2. The message-passing method allows the processes to exchange messages. The responsibility for providing communication may rest with the operating system itself. These two schemes are not mutually exclusive and can be used simultaneously within a single operating system.

8-A Relationship between user threads and kernel threads.

- *Many-to-One*
- *One-to-One*
- *Many-to-Many*

The operating system must select processes from various scheduling queues. Long-term (job) scheduling is the selection of processes that will be allowed to contend for the CPU. Normally, long-term scheduling is heavily influenced by resource-allocation considerations, especially memory management. Short-term (CPU) scheduling is the selection of one process from the ready queue.

a. Operating System Structure
   i. Simple Structure
   ii. Layered Structure
   iii. Microkernel System Structure
   iv. Modular Kernel Structure