**Assignment 1 Set-1**

1. Elaborate System call with example.
2. Elaborate Process states with diagram.
3. Differentiate between Preemptive and non-preemptive algorithms with examples.
4. Elaborate resource allocation graph algorithm
5. Case study of a latest OS.

**Assignment 1 (Set 2)**

1. Elaborate PCB with appropriate diagram.
2. Differentiate between system and application software.
3. Explain the methods of deadlock recovery.
4. In Process management, how Processes are Created and terminated?
5. Why Linux is more secure than any other OS? Explain

**Assignment 1 (Set 3)**

1. Define turnaround time.
2. Differentiate between mutex and semaphore.
3. Differentiate between process and program.
4. Compare ant three types of OS in context to security.
5. Consider the following and find if it is a safe state or not:

Available system resources:

A B C D

Free 3 0 1 2

Processes (currently allocated resources):

A B C D

P1 1 2 2 1

P2 1 1 3 3

P3 1 2 1 0

Processes (maximum resources):

A B C D

P1 3 3 2 2

P2 1 2 3 4

P3 1 3 5 0

**Assignment 1 (Set 4)**

1. Elaborate latency time.
2. Explain the condition required for deadlock.
3. Elaborate the concept of threads.
4. Explain any two scheduling algorithms with example.
5. Explain the concept of safe state with the help of an example.

**Assignment 1 (Set 5)**

1. Elaborate multiprogramming and multi tasking operating system.
2. Elaborate Process states with the help of real life example.
3. Elaborate the bankers algorithm with example.
4. How we can find whether it a safe state of any process.
5. Elaborate Producer Consumer algorithms.

**Assignment 1 (Set 6)**

1. **If we don’t use any operating system, what all will be the challenges faced by the user?**

2. Discuss the **benefits of multi threaded programming.**

3. Consider the following and find if it is a safe state or not:

Available system resources:

A B C D

Free 3 0 1 2

Processes (currently allocated resources):

A B C D

P1 1 2 2 1

P2 1 1 3 3

P3 1 2 1 0

Processes (maximum resources):

A B C D

P1 3 3 2 2

P2 1 2 3 4

P3 1 3 5 0

1. Elaborate dining Philosopher Problem.
2. Elaborate System call with example.

**Assignment 1 (Set 7)**

1. Write some classical problems of Synchronization?
2. How can we determine the safe and unsafe states of a system
3. Explain the necessary conditions that may lead to a deadlock situation.
4. What is a resource allocation graph? How do you obtain a wait-for graph from it? Explain their uses.
5. Can a system detect that some of its processes are starving? If you answer “yes,” explain how it can. If you answer “no,” explain how the system can deal with the starvation problem.

**Assignment 1 (Set 8)**

1. What are the various methods for handling deadlocks?
2. Is it possible to have a deadlock involving only one single process? Explain your answer.
3. Can a system detect that some of its processes are starving? If you answer “yes,” explain how it can. If you answer “no,” explain how the system can deal with the starvation problem.
4. Consider the 4 processes, P1, P2, P3 and P4 shown in the table.

Process Arrival time Burst Time

P1 0 5

P2 1 4

P3 3 7

P4 7 9

Give the completion order of the 4 processes under the policies FCFS. Also find the average waiting time and turnaround time?

1. Differentiate between shared and dedicated devices.

**Assignment 1 (Set 9)**

1. Differentiate between deadlock and starvation.
2. What is critical section problem? What are the requirements that a solution to critical section problem must satisfy?
3. What do you mean by process synchronization? Why is it required?
4. If a system has <A,B,C> three different type of resources & <P1,P2,P3,P4> four different processes having need <2,3>, <4,5>,<3,4> & <4,4>. What will be the minimum number of resources of each type required, so that there will be no deadlock.
5. Differentiate between kernel level thread and user level thread.

**Assignment 1 (Set 10)**

1. If following program executes then, what is the number of child processes created?

For( i=1 ; i<5 ; i++ )

{

Fork();

}

1. .How can we say the First Come First Served scheduling algorithm is Non Pre-emptive?
2. **For what reason Banker’s algorithm is named so? What parameters it use for deadlock avoidance?**
3. Four jobs to be executed on a single processor system arrive at time 0 in the order A, B, C, D. their burst CPU time requirements are 4, 1, 8, 1 time units respectively. What is the completion time of A under round  robin scheduling with time slice of one unit?
4. Differentiate between process and program.

**Assignment 1 (Set 11)**

1. A system has five processes P1 through P5 and four resource types R1 through R4.

There are 2 units of each resource type. Given that:

P1 holds 1 unit of R1 and requests 1 unit of R

P2 holds 1 unit of R3 and requests 1 unit of R2

P3 holds one unit of R2 and requests 1 unit of R3

P4 requests 1 unit of R4

P5 holds one unit of R3 and 1 unit of R2, and requests 1 unit of R3

Show the resource graph for this state of the system. Is the system in deadlock, and if so, which processes are involved?

1. Elaborate the bankers algorithm with example
2. What do you mean by process synchronization? Why is it required?
3. Differentiate between shared and dedicated devices.
4. Explain the condition required for deadlock

**Assignment 1 (Set 12)**

1. Explain different types of System Calls for Windows and Unix systems. Name the system calls used for windows and unix operating system.
2. Is there any solution possible to Non-preemption in deadlock.
3. Differentiate between process and program.
4. Consider a set of n tasks with known runtimes r1, r2, … rn to be run on a uniprocessor machine. Which of the following processor scheduling algorithms will result in the maximum throughput? Explain  
   (a) Round-Robin         (b) Shortest-Job-First  
   (c) Highest-Response-Ratio-Next (d) First-Come-First-Served
5. A process executes the code

fork();

fork();

fork();

What is the total number of child processes created?