Assignment Questions
CS204 OPERATING SYSTEMS

Question Number	Roll Numbers	Date of Submission
1	1,18,35,52	12/04/2019
2	2,19,36,53	
3	3,20,37,54	
4	4,21,38,55	
5	5,22,39,56	
6	6,23,40,57	
7	7,24,41,58	
8	8,25,42,59	
9	9,26,43,60	
10	10,27,44,61	
11	11,28,45,62	
12	12,29,46,63	
13	13,30,47,64	
14	14,31,48,65	
15	15,32,49,66	
16	16,33,50,67	
17	17,34,51	

1. Race conditions are possible in many systems. Consider a banking system that maintains an account balance with two functions: deposit(amount) and withdraw(amount). These two functions are passed the amount that is to be deposited or withdrawn from the bank account balance. Assume that a husband and wife share a bank account. Concurrently, the husband calls the withdraw() function and the wife calls deposit() . Describe how a race condition is possible and what might be done to prevent the race condition from occurring.
2. Consider how to implement a mutex lock using an atomic hardware instruction. Assume that the following structure defining the mutex lock is available:
   ```
   typedef struct {
           int available;
       } lock ;
   ```
   (available == 0) indicates that the lock is available, and a value of 1 indicates that the lock is unavailable. Using this struct, illustrate how the following functions can be implemented using the test and set() instructions:
   • void acquire(lock *mutex)
   • void release(lock *mutex)
   Be sure to include any initialization that may be necessary.
3. Servers can be designed to limit the number of open connections. For example, a server may wish to have only N socket connections at any point in time. As soon as N connections are made, the server will

not accept another incoming connection until an existing connection is released. Explain how semaphores can be used by a server to limit the number of concurrent connections.

4. A file is to be shared among different processes, each of which has a unique number. The file can be accessed simultaneously by several processes, subject to the following constraint: the sum of all unique numbers associated with all the processes currently accessing the file must be less than n. Write a monitor to coordinate access to the file.
5. Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P 1	2	2
P 2	1	1
P 3	8	4
P 4	4	2
P 5	5	3

The processes are assumed to have arrived in the order P 1, P 2, P 3, P 4, P 5, all at time 0.

a. Draw Gantt charts that illustrate the execution of these processes using the FCFS scheduling algorithms

b. What is the turnaround time of each process.

c. What is the waiting time of each process.

d. Find the minimum average waiting time.

6. Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P 1	2	2
P 2	1	1
P 3	8	4
P 4	4	2
P 5	5	3

The processes are assumed to have arrived in the order P 1 , P 2 , P 3 , P 4 , P 5 , all at time 0.

a. Draw Gantt charts that illustrate the execution of these processes using the SJF scheduling algorithm.

b. What is the turnaround time of each process.

c. What is the waiting time of each process.

d. Find the minimum average waiting time.

7. Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P 1	2	2
P 2	1	1
P 3	8	4
P 4	4	2
P 5	5	3

The processes are assumed to have arrived in the order P 1 , P 2 , P 3 , P 4 , P 5 , all at time 0.

a. Draw Gantt charts that illustrate the execution of these processes using the non-preemptive priority (a larger priority number implies a higher priority) scheduling algorithm.

b. What is the turnaround time of each process.

c. What is the waiting time of each process.

d. Find the minimum average waiting time.

8. Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P 1	2	2
P 2	1	1

P 3	8	4
P 4	4	2
P 5	5	3

The processes are assumed to have arrived in the order P 1 , P 2 , P 3 , P 4 , P 5 , all at time 0.

a. Draw Gantt charts that illustrate the execution of these processes using the RR (quantum = 2). scheduling algorithm.

b. What is the turnaround time of each process.

c. What is the waiting time of each process.

d. Find the minimum average waiting time.

9. Consider the following snapshot of a system:

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P0	0 0 1 2	0 0 1 2	1 5 2 0
P1	1 0 0 0	1 7 5 0	
P2	1 3 5 4	2 3 5 6	
P3	0 6 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	

Answer the following questions using the banker's algorithm:

a. What is the content of the matrix Need?

b. Is the system in a safe state?

c. If a request from process P 1 arrives for (0,4,2,0), can the request be granted immediately?

10. Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free.

11. Consider the following snapshot of a system:

Process	Allocation	Max
	A B C D	A B C D
P0	3 0 1 4	5 1 1 7
P1	2 2 1 0	3 2 1 1
P2	3 1 2 1	3 3 2 1
P3	0 5 1 0	4 6 1 2
P4	4 2 1 2	6 3 2 5

Using the banker's algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the processes may complete. Otherwise, illustrate why the state is unsafe.

a. Available = (0, 3, 0, 1)

b. Available = (1, 0, 0, 2)

12. Implement an Airline reservation system using monitors. The system must process booking and cancellation of requests and perform appropriate actions. Identify the shared data, operations and processes required.

13. The synchronization problem called *sleeping barber* is defined as follows: A barber shop has a single barber, a single barber's chair in a small room and a large waiting room with n seats. The barber and his chair are visible from the waiting room. After servicing one customer, the barber checks if any customers are waiting in the waiting rooms. If so, he admits one of them and starts serving him, else he goes to sleep in the barber's chair. A customer enters the waiting room only if there is at least one vacant seat and either waits for the barber to call him if the barber is busy or wakes the barber if he is asleep.

Identify the synchronization requirements between the barber and customer processes. Code the barber and customer process such that deadlocks do not arise.

14. A system containing preemptible resources uses the following resource allocation policy: When a resource requested by some process Pi is unavailable

a) The resource is preempted from one of its holder processes Pj, if Pj is younger than Pi. The resource is now allocated to Pi. It is allocated back to Pj when Pi completes. (A process is considered to be younger if it was initiated later.)

b) If the condition in (a) is not satisfied, Pi is blocked for the resources.

A released resource is always allocated to its oldest requester. Prove that deadlocks cannot arise in this system. Also prove that each process completes its execution in finite time.

15. In the following system:

	Allocation	Max	Available	Total
	A B C	A B C	A B C	A B C
P0	3 6 8	2 2 3	5 4 10	7 7 10
P1	4 3 3	2 0 3		
P2	3 4 4	1 2 4		

(a) Is the current allocation state safe?

(b) Would the following requests be granted in the current state?

(i) Process P0 requests (1,1,0)

(ii) Process P2 requests (0,1,0)

(iii) Process P1 requests (0,1,0)

16. A road crosses a railway track at two points. Gates are constructed on the road at each crossing to stop road traffic when a train is about to pass. Train traffic is stopped if a car blocks the track. Two-way traffic of cars is permitted on the road and two-way train traffic is permitted on the railway tracks.

(a) Discuss whether deadlocks can arise in the road and train traffic. Would there be no deadlocks if both road and train traffic are only one way?

(b) Design a set of simple rules to avoid deadlocks in the road_and_train traffic

17. A system is composed of four processes, P0, P1, P2, P3, three types of resources, R0, R1, R2. The number of resources are: <3,2,2>

Process P0 holds one unit of R0 and requests one units of R1.

P1 holds two units of R1 and requests one unit each of R0 and R2

P2 holds one unit of R0 and requests one unit of R1.

P3 holds two units of R2 and requests one unit of R0.

Draw the Resource Allocation Graph to represent the system state and check whether there can be a deadlock?