***Jaypee Institute of Information Technology, Noida***

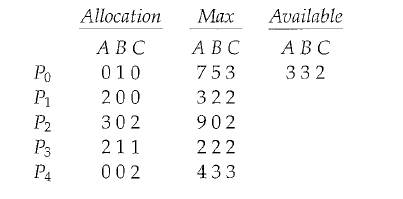
***Operating System and System Programming lab [15B17CI472]***

***Assignment- 7***

1. To avoid deadlock in dining philosophers’ problem use a possible solution as the odd numbered philosophers grab the right and then the left. Implement this solution using pthread mutual exclusion lock.
2. Using condition variables to implement a producer-consumer algorithm. Define two threads: one producer and one consumer. The producer reads characters one by one from a string stored in a file named “string.txt”, then writes sequentially these characters into a circular queue. Meanwhile, the consumer reads sequentially from the queue and prints them in the same order. The diagram illustrates the process. Upon completion of running the program, “Hello! World.” is printed on the screen. In the program, use #define to specify the size of the queue. For example, #define QUEUE\_SIZE 5. Make sure to test your program with different queue sizes, including 1.



1. Write a program to implement producer consumer problem (Using MUTEX semaphores).
2. Consider a system with: five processes, P0- P4, three resource types, A, B, and C. Type A has 10 instances, B has 5 instances, C has 7 instances. At time T0 the following snapshot of the system is taken.



Write a Linux C program to check weather system is in safe state or not. Also demonstrate the unsafe sequence by modifying any resource allocation.

1. Write a Linux C program to demonstrate that deadlock and starvation are opposite problems. Solution to deadlock problem cause more starvation and solution to the starvation problem can cause deadlock.
2. The barber has one barber chair and a waiting room with a number of chairs in it. When the barber finishes cutting a customer's hair, he dismisses the customer and then goes to the waiting room to see if there are other customers waiting. If there are, he brings one of them back to the chair and cuts his hair. If there are no other customers waiting, he returns to his chair and sleeps in it.

Each customer, when he arrives, looks to see what the barber is doing. If the barber is sleeping, then the customer wakes him up and sits in the chair. If the barber is cutting hair, then the customer goes to the waiting room. If there is a free chair in the waiting room, the customer sits in it and waits his turn. If there is no free chair, then the customer leaves. Based on a naïve analysis, the above description should ensure that the shop functions correctly, with the barber cutting the hair of anyone who arrives until there are no more customers, and then sleeping until the next customer arrives.

Solve the above problem using conditional semaphore.