National University of Computer and Emerging Sciences

Laboratory Manual

(Operating Systems)

Semester	Spring 2021

Department of Computer Science

FAST-NU, Lahore

Objective:

To learn synchronization through Semaphores between threads

Question # 1

There are exactly 3 threads generate string a, b and c in an arbitrary order. In an absence of any synchronization mechanism there will be no order in generation of a, b and c. Synchronize threads using semaphore in such a way that your printed string will be aaacbaaacbaaacb……

//thread 1
While(1)
{
 	
cout << ‘a’;
 	
}
	//thread 2
	
	While(1)
	{
	
	Cout << ‘b’;
	
	}
		//thread 3
		
		While(1)
		{
		
		Cout << ‘c’;
		
		}

Question # 2

We have two kinds of threads, the writer thread and the reader thread. The writer thread asks the user for input, and writes the input to shared memory. The reader thread then reads the data from shared memory and prints the data on the screen. Write a program which has 1 writer thread and n reader threads (n is a parameter asked from user). The program will do the following:

1. The writer takes data from the user and writes it to shared memory.
2. N readers then read the data written to the shared memory by the writer. However, all readers will not read in parallel, they will read the data from shared memory one after the other. For example if there are 3 threads, then thread 1 will read the data first. While thread 1 is reading no other thread is allowed to read. After thread 1 has finished reading, thread 2 will read. After thread 2 has finished reading, thread 3 will read. The other order in which threads read does not matter.
3. After the completion of step 2, step 1 will execute again. (So, there is a loop in writer and reader threads, and they run infinitely).