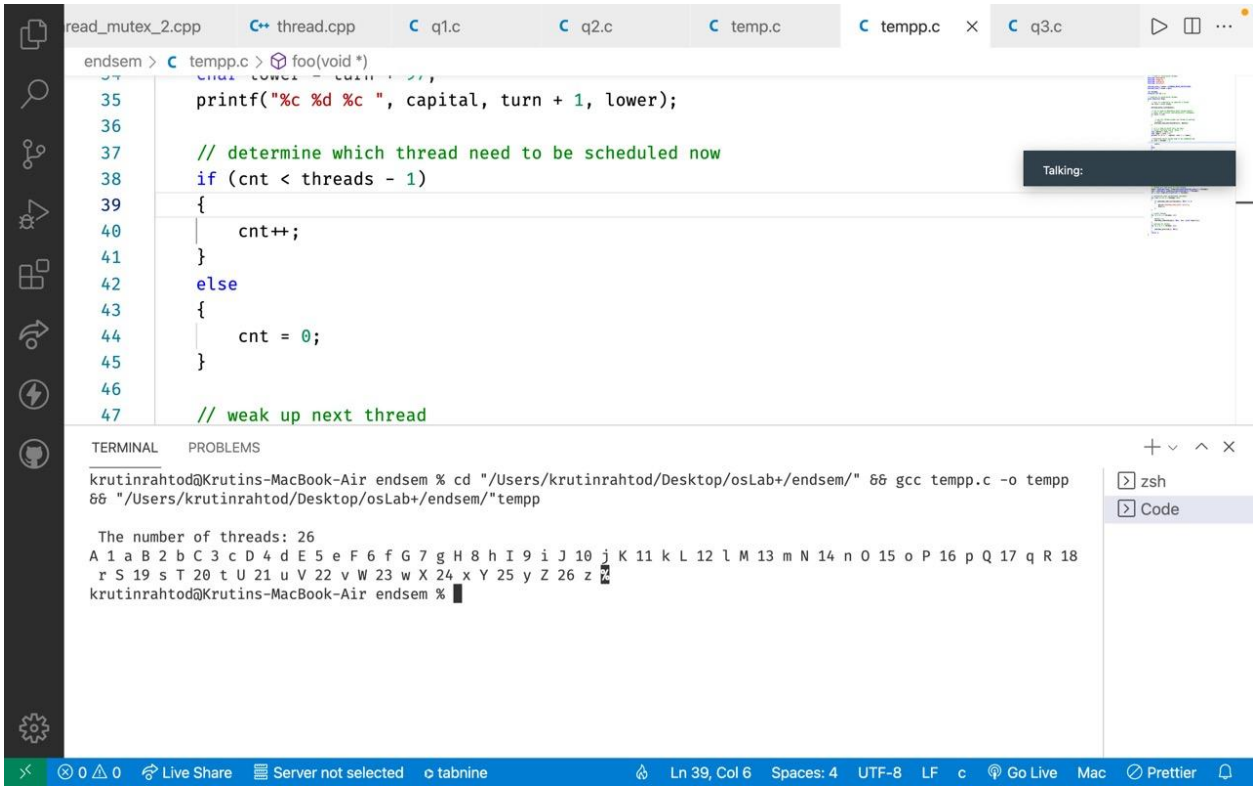
OS End Sem lab exam
Name : Krutin Rathod
Roll No : AU1940261

Question - 1

First of all we need to include pthread.h in order to work with threads or perform any task. F is used to identify a thread. currentThread is taken to determine the thread which will enter into the critical section. pthread_create() function will create a thread. pthread_create() will take 4 arguments. The first one is a pointer to thread_id which is set by this function. The second will specify the attributes. In case of null default value will be used. The third argument is name of function to be executed for the thread to be created. The fourth argument is used to pass arguments to the function, function.

The pthread_join() function for threads is the equivalent of wait() for processes. A call to pthread_join blocks the calling thread until the thread with identifier equal to the first argument terminates.

The overall idea is to generate 3 types of threads. One for capital letters, second for numbers and third for lowercase letters. We will then use thread synchronization to produce the required output. We have created a function to synchronize threads by name function. Depending upon the thread we will name the turn accordingly of the thread and thus produce the output. On count == 0 first thread will print A, second thread will print 2 and thus will loop over 26 times to display the desired output.



Question - 2

Round robin is a preemptive algorithm that means it will change the current task after a certain time quantum. We add a new process in queue, if we are performing a task means a time quantum is running. After completion of a time quantum we will check the ready queue, then from that process having lowest burst time is taken. After each time quantum a new process is taken by the CPU. Performance of RR depends on the time quantum.

The difference between MRR and RR is Average waiting time and Average Turn around time both are less. In that algorithm also the turnaround time is less. Thus better performance. Therefore, it overcomes all the drawbacks of the RR algorithm as the proposed algorithm has less no. of switches, less amount of turn around as well as waiting time.

Question - 3

In producer consumer problem producers are generating certain types of data and putting them in a place called buffer. buffer may be of an infinite or finite length. Consumers are the ones who are taking data out of the buffer. They are taking out data one by one. We also need to take care that when the buffer is full, producer do not add more data, and also when the consumer is empty it should not try to remove the data from the empty buffer. To solve the producer consumer problem, binary semaphores are used. Firstly we need 2 semaphores namely full and empty, which keeps track of no. of full and empty slots in the buffer.

In consumer's code we will down the full as it consumes the data item. Value of binary semaphore s or mutex is reduced so that the producer cannot activate the critical section right now, then after completing the task consumer will increase the mutex and also the empty.

In the producer's code the reverse will happen. It will down the empty and also the mutex, so consumer cannot preempt the process. After completing his process, the producer will increase the mutex as well as full.