**Name : Lankadasu Naga Venkata Yaswanth**

**Reg. No. : 12104843**

**Email Address : lnvyeswanth1575@gmail.com**

**GITHUB Link :** <https://github.com/yaswanth-github/CSE316>

**Question : 1**

There are 3 student processes and 1 teacher process. Students are supposed to do their assignments and they need 3 things for that-pen, paper and question paper. The teacher has an infinite supply of all the three things. One student has pen, another has paper and another has question paper. The teacher places two things on a shared table and the student having the third complementary thing makes the assignment and tells the teacher on completion. The teacher then places another two things out of the three and again the student having the third thing makes the assignment and tells the teacher on completion. This cycle continues. WAP to synchronise the teacher and the students.

• Two types of people can enter into a library- students and teachers. After entering the library, the visitor searches for the required books and gets them. In order to get them issued, he goes to the single CPU which is there to process the issuing of books. Two types of queues are there at the counter-one for students and one for teachers. A student goes and stands at the tail of the queue for students and similarly the teacher goes and stands at the tail of the queue for teachers (FIFO). If a student is being serviced and a teacher arrives at the counter, he would be the next person to get service (PRIORITY-non preemptive). If two teachers arrive at the same time, they will stand in their queue to get service (FIFO). WAP to ensure that the system works in a non-chaotic manner.

• If a teacher is being served and during the period when he is being served, another teacher comes, then that teacher would get the service next. This process might continue leading to increase in waiting time of students. Ensure in your program that the waiting time of students is minimized.

**Description:**

* This is a scheduling program that is capable of scheduling many processes that comes in at some time interval and is allocated the CPU.
* This scheduler which schedules the job by considering the arrival time of the processes where arrival time if given as 0 is discarded or displayed as error.
* The scheduler implements the shortest job first scheduling policy, but checks the queue of the processes after the every process terminates and time taken for checking and arranging the process according to the shortest job is 2 time unit.
* At last it computes the waiting time for process, turnaround time for process and average waiting time and turnaround time and the total time taken by the processor to compute all the jobs.

**Synchronizing the teacher and students:**

In this problem, we need to synchronize the teacher and the students in order to ensure that each student has all the required items to complete their assignment.

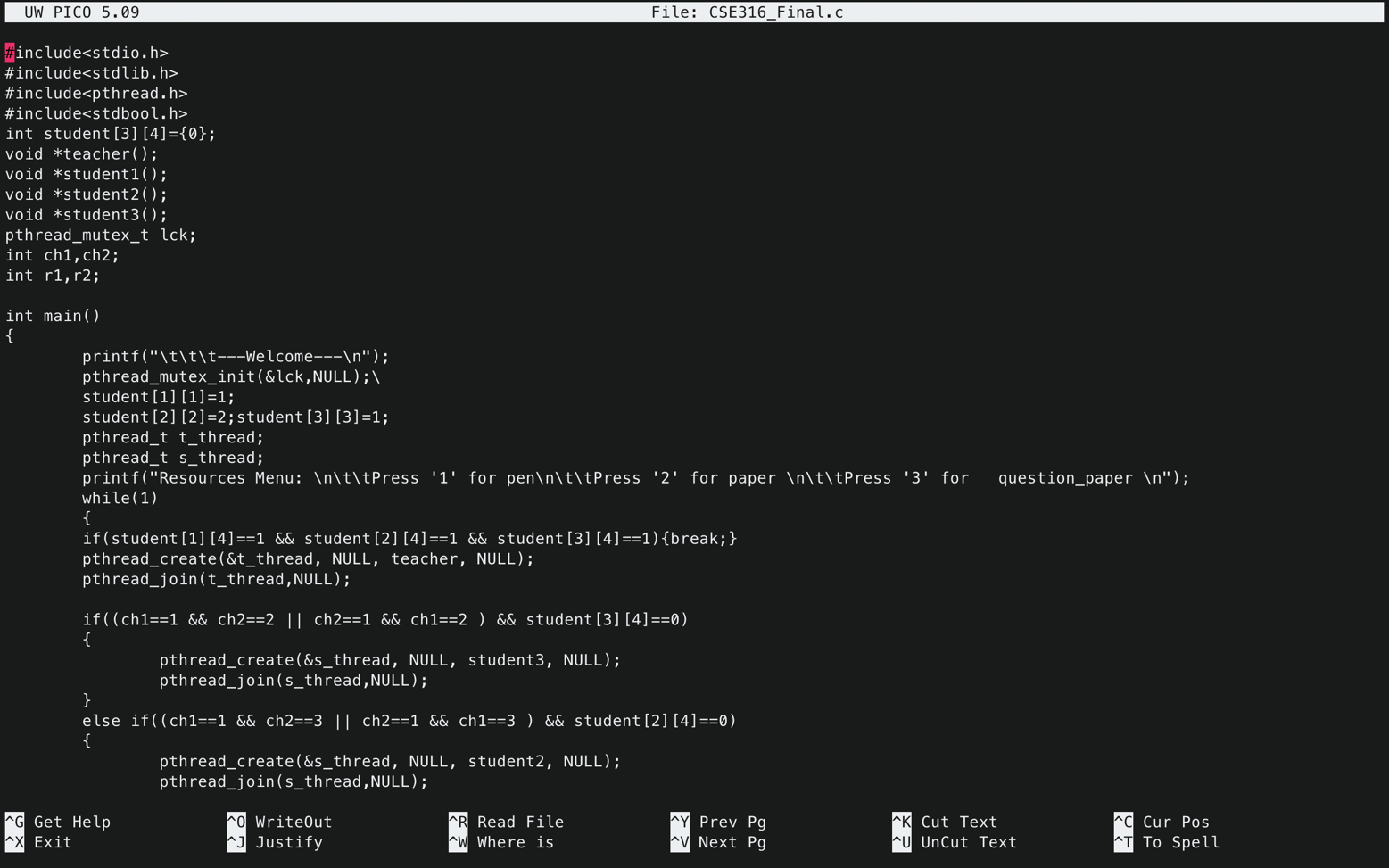
we can use semaphores to synchronize the teacher and students. We can use three semaphores - one for each item (pen, paper, question paper). Initially, all three semaphores are set to 1. When a student needs an item, he waits on the semaphore for that item. When the teacher places an item on the table, he signals the semaphore for that item. This way, only one student can access an item at a time.

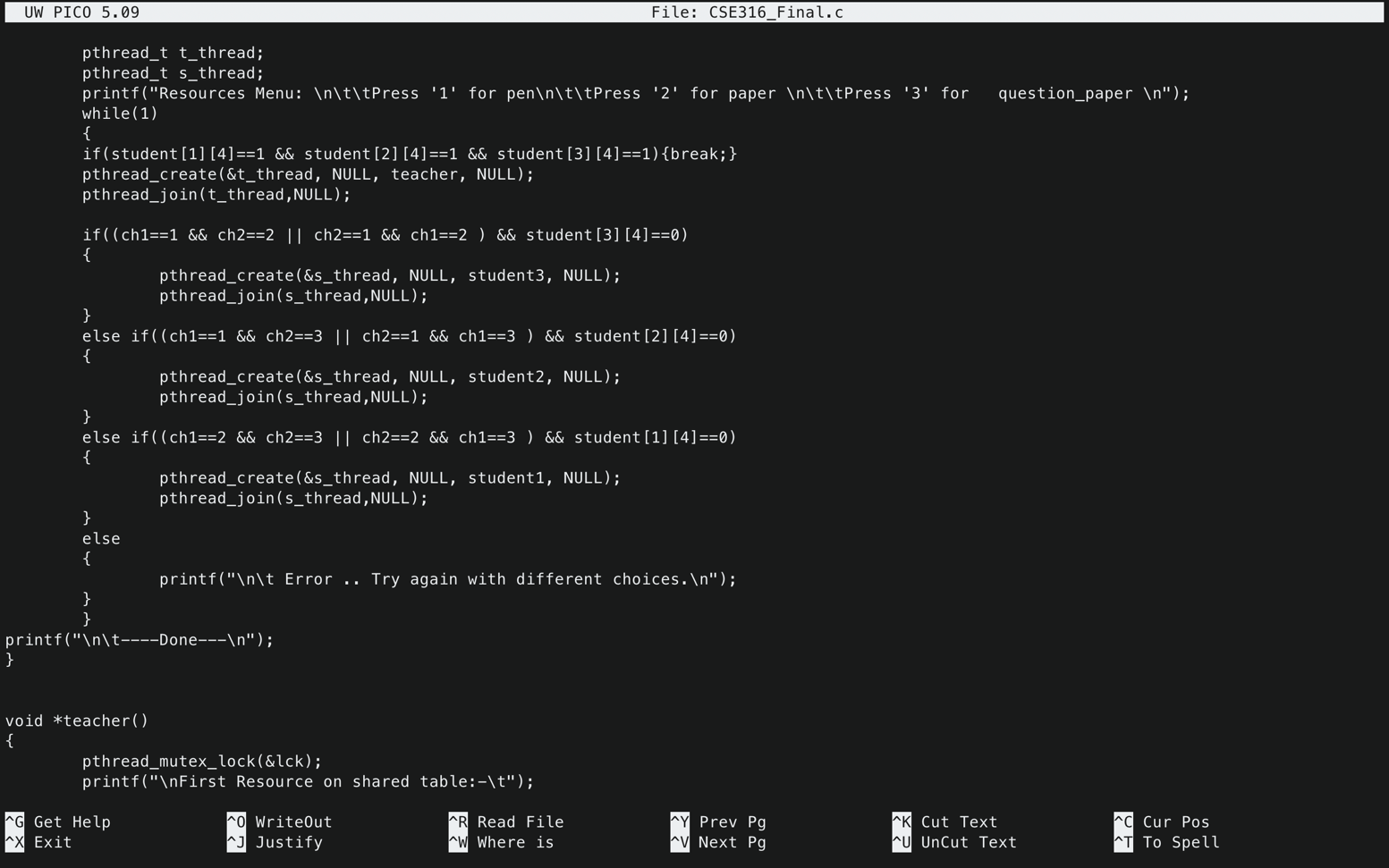
**Minimizing waiting time of students:**

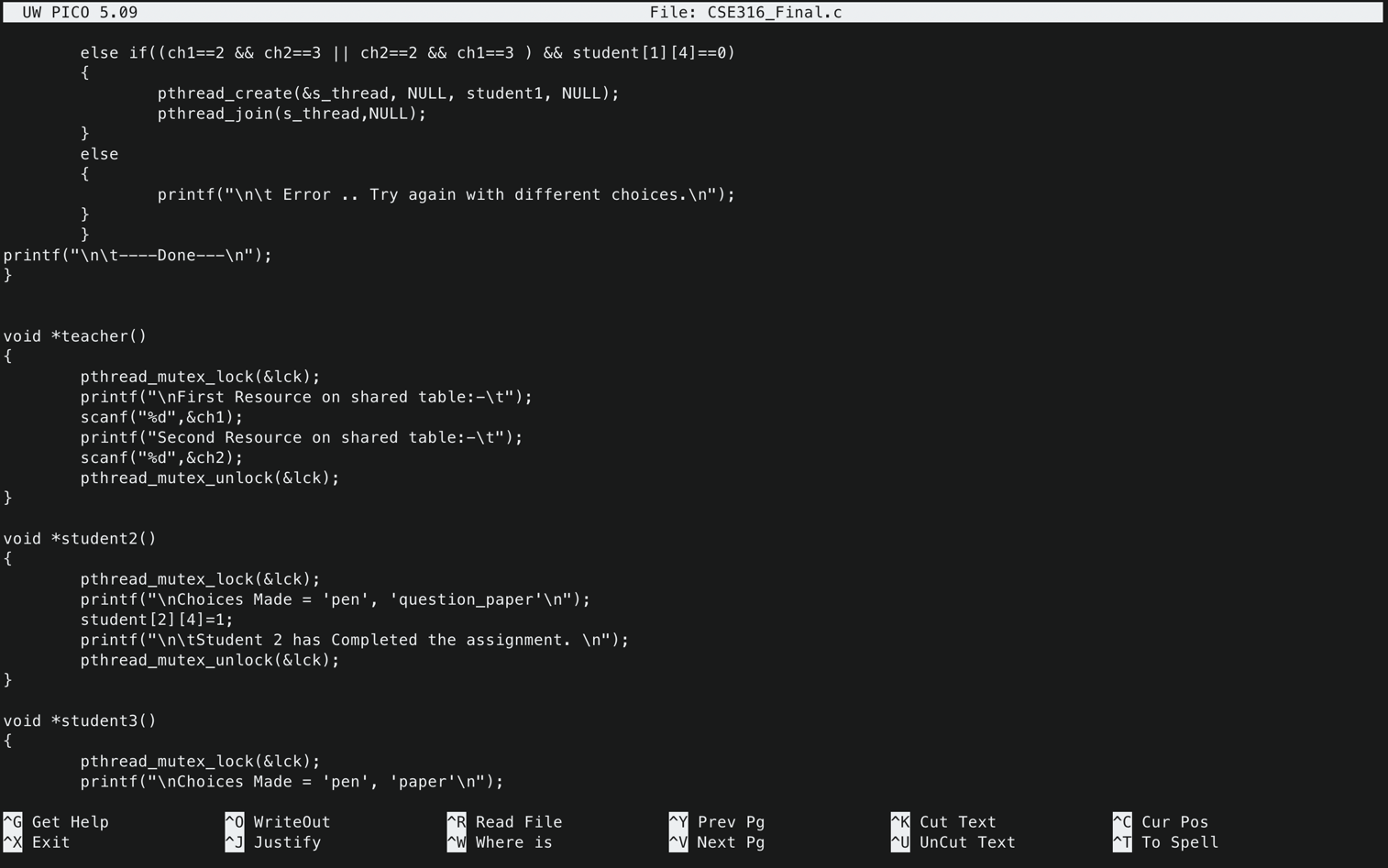
To minimize waiting time of students when a teacher is being served and another teacher comes during that period, we can use another semaphore to keep track of how many teachers are being served. When a teacher arrives at the counter and there are no other teachers being served, he is given priority. If there are other teachers being served, he waits on this semaphore until all other teachers have been serviced.

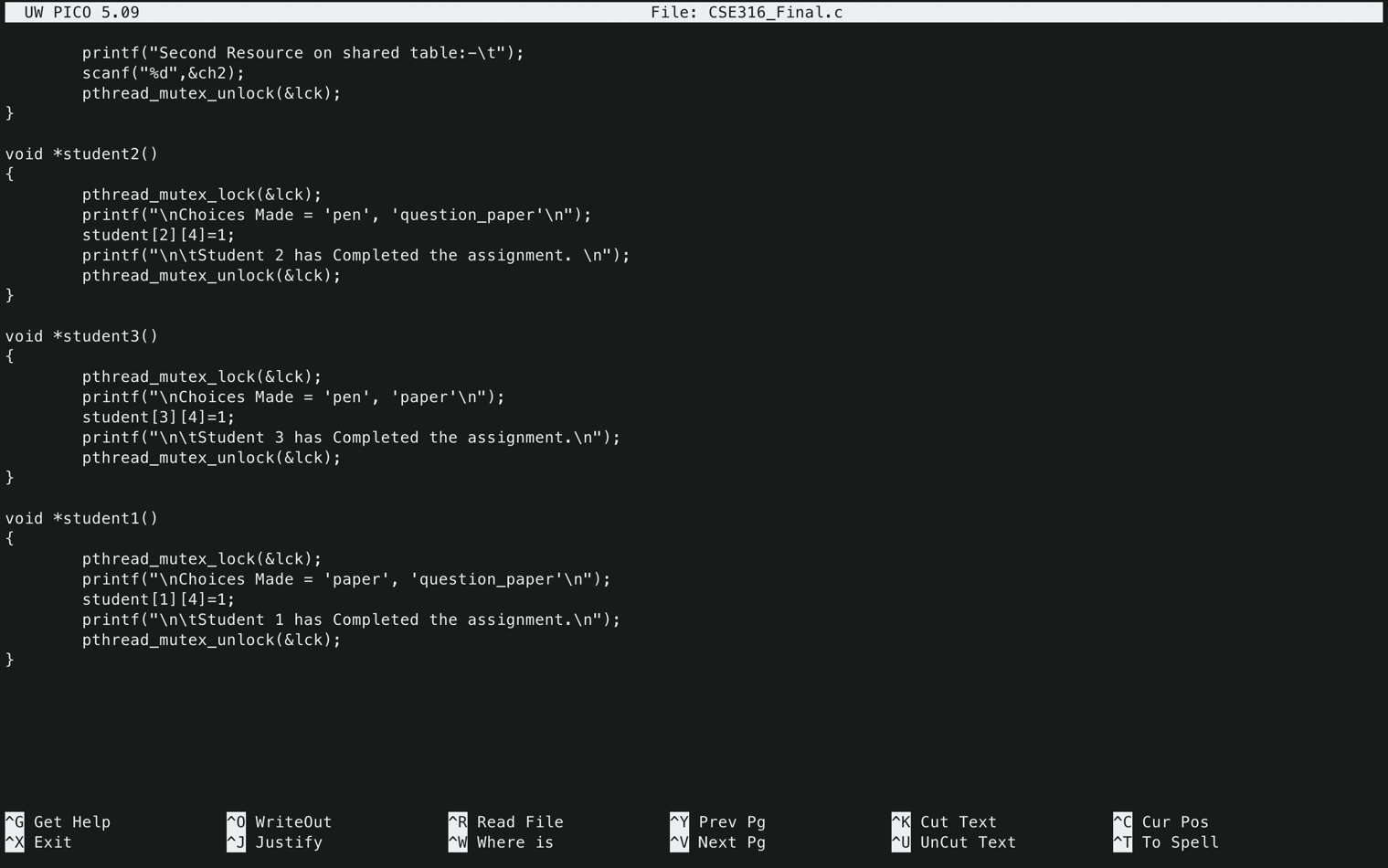
**Methodology :**

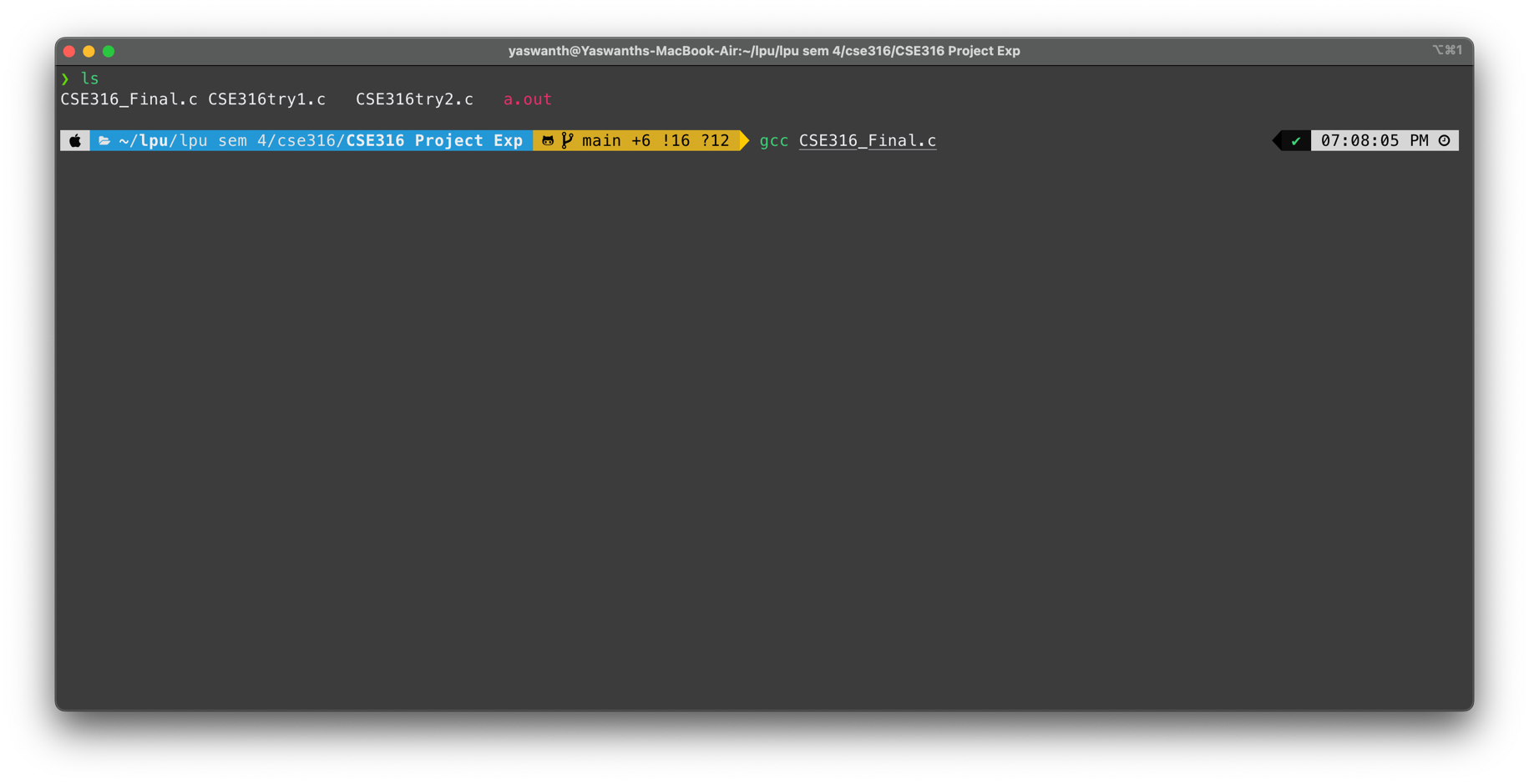
To solve this problem, we will use a priority queue to prioritize teachers over students. When a teacher arrives while a student is being serviced, the teacher will be placed at the head of the teacher queue. When the student is done being serviced, the CPU process will first check the teacher queue before serving the next person in the student queue.

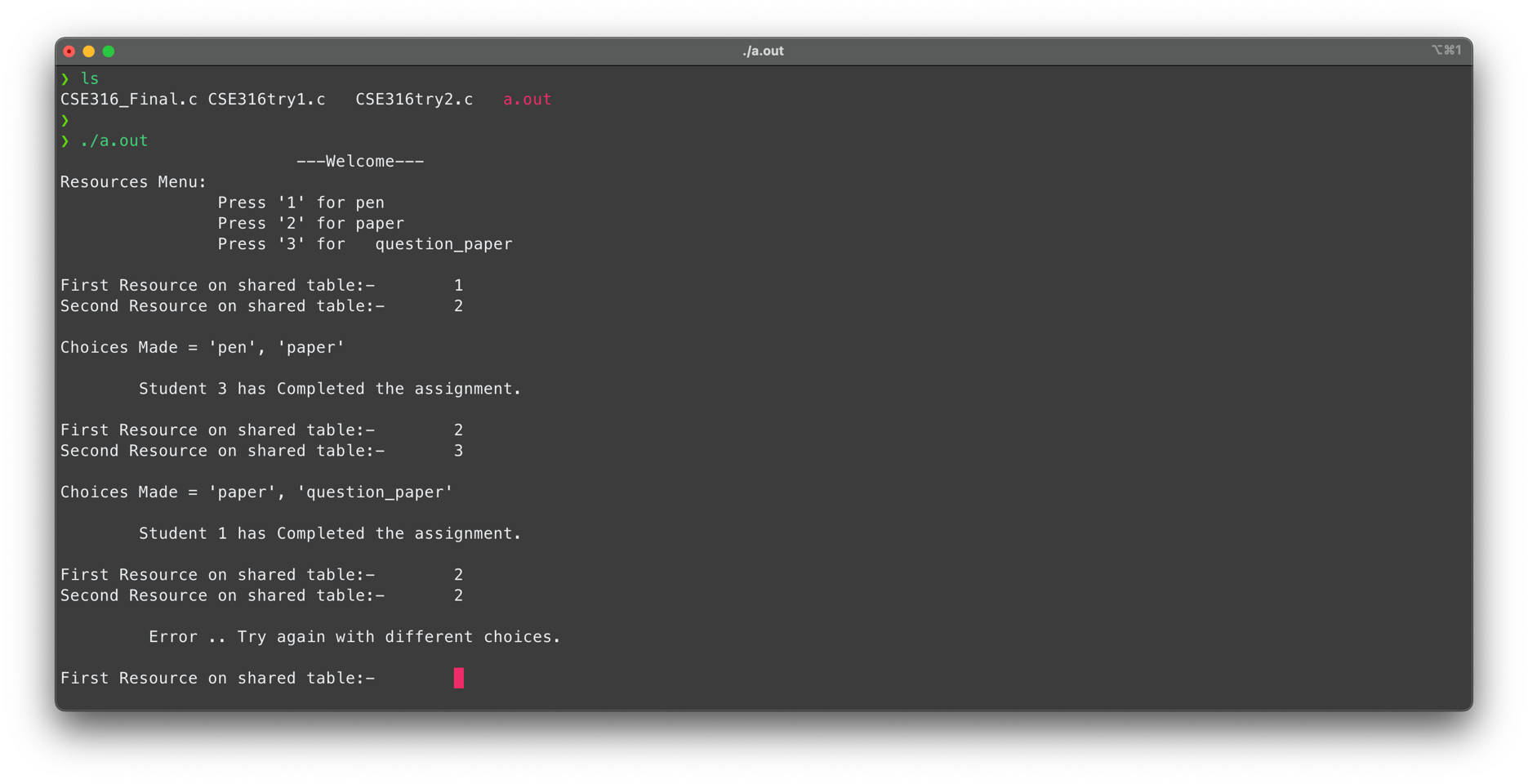
**Code snippet:**

****

****

****

**GCC**

**Outputs**

****

**Test Cases :GitHub Link**:

The following code has been uploaded to the GitHub by making a repository named “CSE316” and the Three revisions were completed.

1st : [CSE316/CSE316try1.c at main · yaswanth-github/CSE316](https://github.com/yaswanth-github/CSE316/blob/main/CSE316try1.c)

2nd: [CSE316/CSE316try2.c at main · yaswanth-github/CSE316](https://github.com/yaswanth-github/CSE316/blob/main/CSE316try2.c)

Final: [CSE316/CSE316\_Final.c at main · yaswanth-github/CSE316](https://github.com/yaswanth-github/CSE316/blob/main/CSE316_Final.c)

=====================================================================