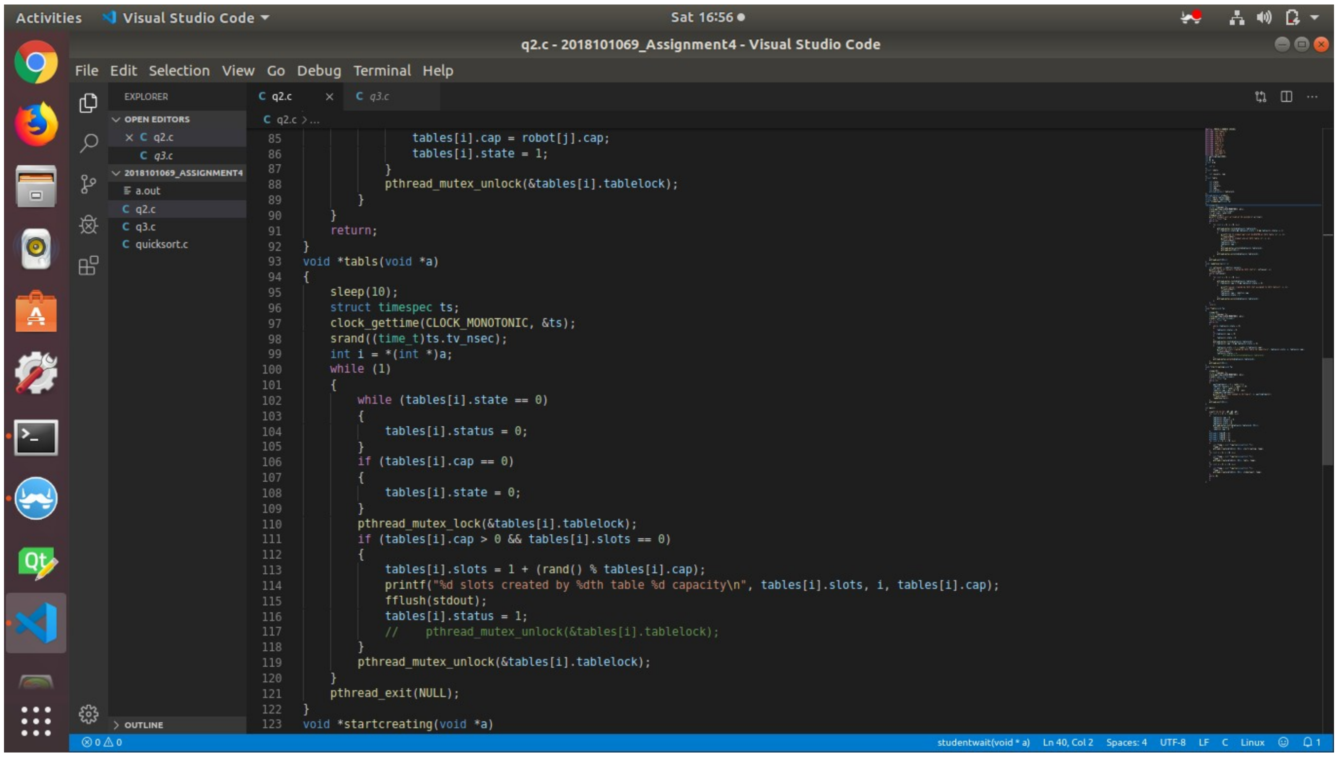
# Report for question 2

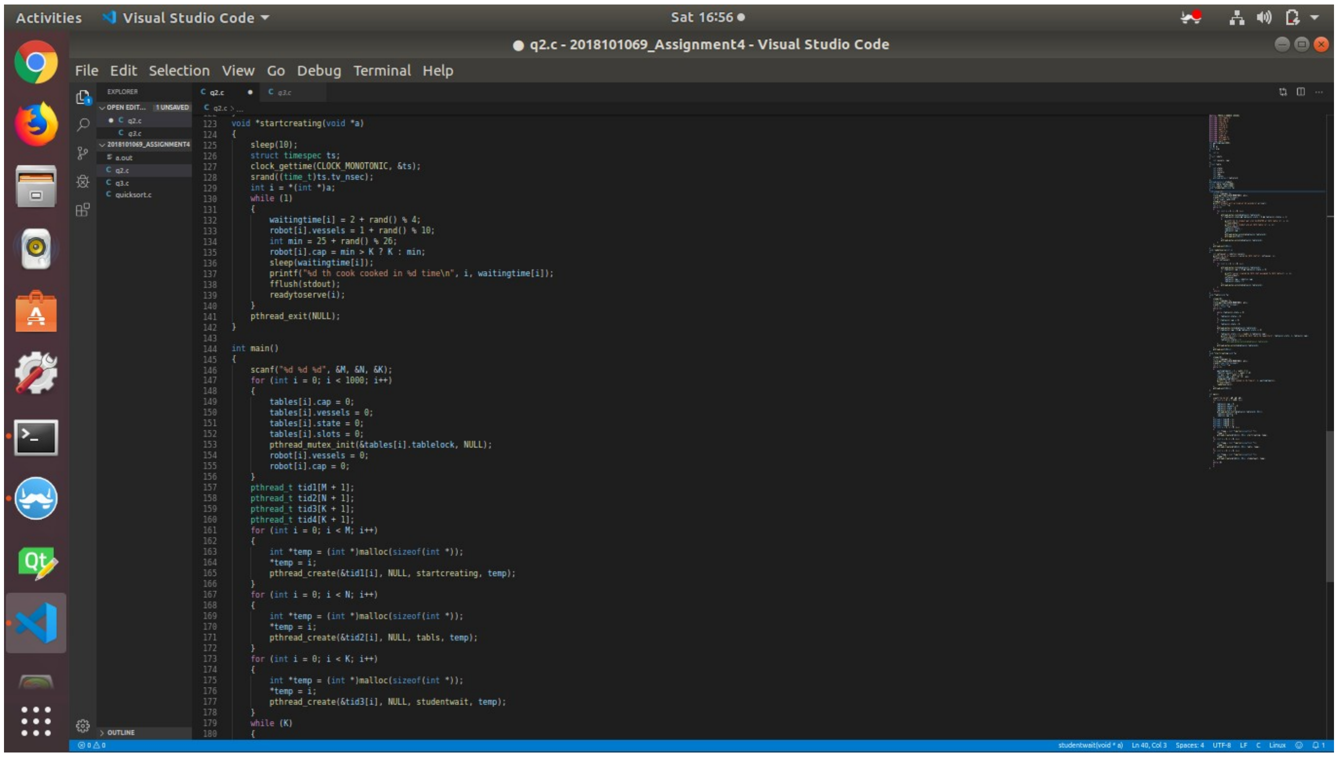


```c
                tables[i].cap = robot[j].cap;
                tables[i].state = 1;
            }
            pthread_mutex_unlock(&tables[i].tablelock);
        }
    }
    return;
}
void *tabls(void *a)
{
    sleep(10);
    struct timespec ts;
    clock_gettime(CLOCK_MONOTONIC, &ts);
    srand((time_t)ts.tv_nsec);
    int i = *(int *)a;
    while (1)
    {
        while (tables[i].state == 0)
        {
            tables[i].status = 0;
        }
        if (tables[i].cap == 0)
        {
            tables[i].state = 0;
        }
        pthread_mutex_lock(&tables[i].tablelock);
        if (tables[i].cap > 0 && tables[i].slots == 0)
        {
            tables[i].slots = 1 + (rand() % tables[i].cap);
            printf("%d slots created by %dth table %d capacity\n", tables[i].slots, i, tables[i].cap);
            fflush(stdout);
            tables[i].status = 1;
            //    pthread_mutex_unlock(&tables[i].tablelock);
        }
        pthread_mutex_unlock(&tables[i].tablelock);
    }
    pthread_exit(NULL);
}
void *startcreating(void *a)
```

Activities
Visual Studio Code
Sat 16:56
q2.c - 2018101069_Assignment4 - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
EXPLORER
OPEN EDITORS
q2.c
q3.c
2018101069_ASSIGNMENT4
a.out
q2.c
q3.c
quicksort.c
OUTLINE
studentwait(void * a) Ln 40, Col 2 Spaces: 4 UTF-8 LF C Linux

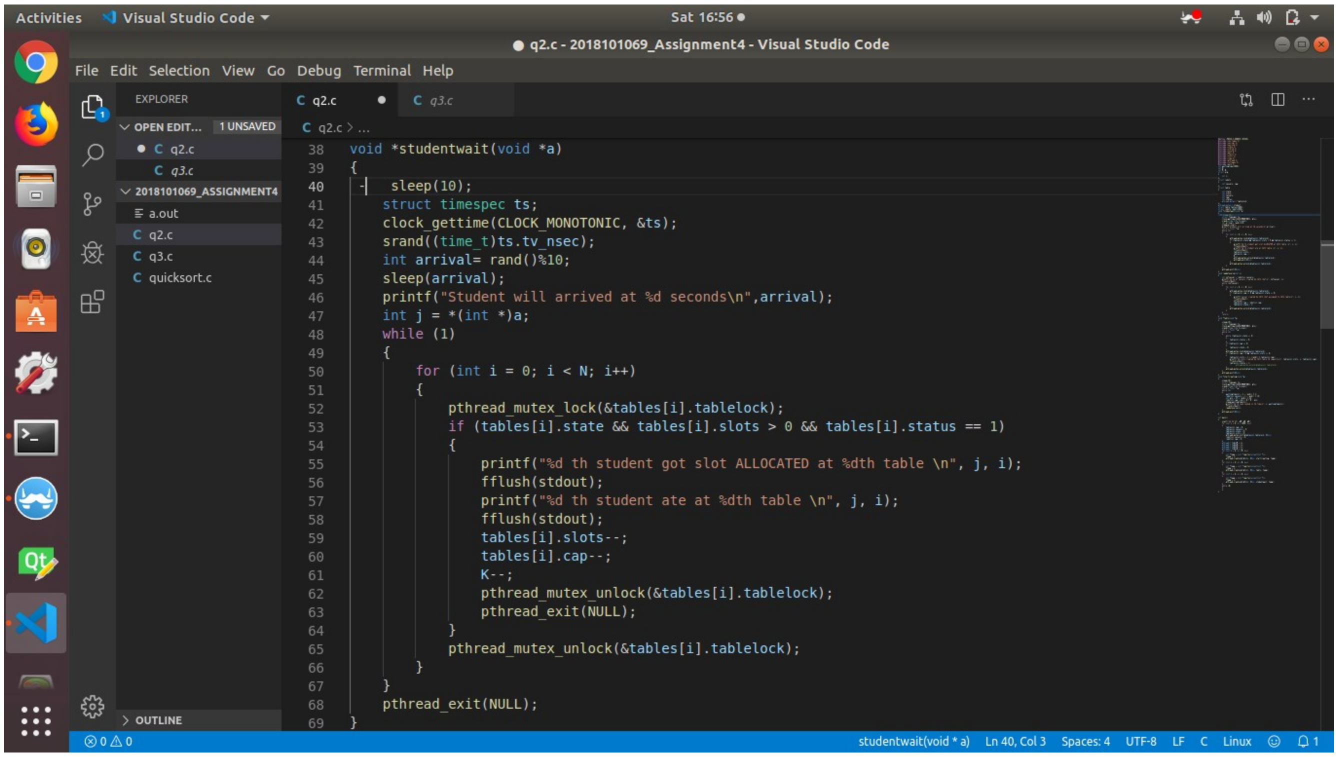
Tabls function is function used for threads of tables and creating slots in the table and changing state of table struct

Startcreating function is function used for threadsof chefs and is used to call readytoserve function when a chef is finished cooking biryani



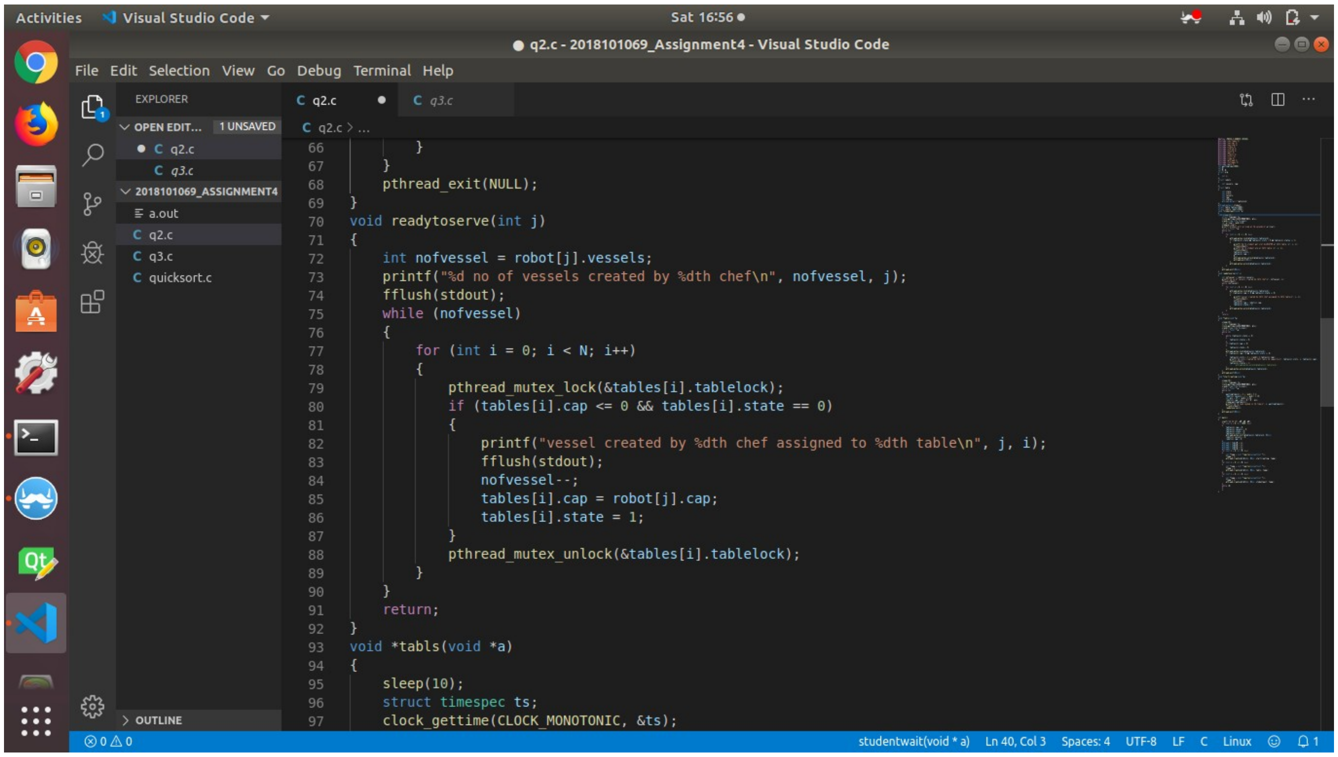
```c
void *startcreating(void *a)
{
    sleep(10);
    struct timespec ts;
    clock_gettime(CLOCK_MONOTONIC, &ts);
    srand((time_t)ts.tv_nsec);
    int i = *(int *)a;
    while (1)
    {
        waitingtime[i] = 2 + rand() % 4;
        robot[i].vessels = 1 + rand() % 10;
        int min = 25 + rand() % 26;
        robot[i].cap = min > K ? K : min;
        sleep(waitingtime[i]);
        printf("%d th cook cooked in %d time\n", i, waitingtime[i]);
        fflush(stdout);
        readytoserve(i);
    }
    pthread_exit(NULL);
}

int main()
{
    scanf("%d %d %d", &M, &N, &K);
    for (int i = 0; i < 1000; i++)
    {
        tables[i].cap = 0;
        tables[i].vessels = 0;
        tables[i].state = 0;
        tables[i].slots = 0;
        pthread_mutex_init(&tables[i].tablelock, NULL);
        robot[i].vessels = 0;
        robot[i].cap = 0;
    }
    pthread_t tid1[M + 1];
    pthread_t tid2[N + 1];
    pthread_t tid3[K + 1];
    pthread_t tid4[K + 1];
    for (int i = 0; i < M; i++)
    {
        int *temp = (int *)malloc(sizeof(int *));
        *temp = i;
        pthread_create(&tid1[i], NULL, startcreating, temp);
    }
    for (int i = 0; i < N; i++)
    {
        int *temp = (int *)malloc(sizeof(int *));
        *temp = i;
        pthread_create(&tid2[i], NULL, tabls, temp);
    }
    for (int i = 0; i < K; i++)
    {
        int *temp = (int *)malloc(sizeof(int *));
        *temp = i;
        pthread_create(&tid3[i], NULL, studentwait, temp);
    }
    while (K)
    {
```

This snippet also shows Main function which is used to create threads for chefs, students and tables. It also uses a while loop to make sure all students gets Biryani.



```c
void *studentwait(void *a)
{
    sleep(10);
    struct timespec ts;
    clock_gettime(CLOCK_MONOTONIC, &ts);
    srand((time_t)ts.tv_nsec);
    int arrival= rand()%10;
    sleep(arrival);
    printf("Student will arrived at %d seconds\n",arrival);
    int j = *(int *)a;
    while (1)
    {
        for (int i = 0; i < N; i++)
        {
            pthread_mutex_lock(&tables[i].tablelock);
            if (tables[i].state && tables[i].slots > 0 && tables[i].status == 1)
            {
                printf("%d th student got slot ALLOCATED at %dth table \n", j, i);
                fflush(stdout);
                printf("%d th student ate at %dth table \n", j, i);
                fflush(stdout);
                tables[i].slots--;
                tables[i].cap--;
                K--;
                pthread_mutex_unlock(&tables[i].tablelock);
                pthread_exit(NULL);
            }
            pthread_mutex_unlock(&tables[i].tablelock);
        }
    }
    pthread_exit(NULL);
}
```

This snippet shows  code for threads of student which is used to get students slots and feed students


Activities
Visual Studio Code
Sat 16:56
● q2.c - 2018101069_Assignment4 - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
EXPLORER
OPEN EDIT... 1 UNSAVED
q2.c
q3.c
2018101069_ASSIGNMENT4
a.out
q2.c
q3.c
quicksort.c
OUTLINE
q2.c > ...
studentwait(void * a) Ln 40, Col 3 Spaces: 4 UTF-8 LF C Linux

```c
        }
    }
    pthread_exit(NULL);
}
void readytoserve(int j)
{
    int nofvessel = robot[j].vessels;
    printf("%d no of vessels created by %dth chef\n", nofvessel, j);
    fflush(stdout);
    while (nofvessel)
    {
        for (int i = 0; i < N; i++)
        {
            pthread_mutex_lock(&tables[i].tablelock);
            if (tables[i].cap <= 0 && tables[i].state == 0)
            {
                printf("vessel created by %dth chef assigned to %dth table\n", j, i);
                fflush(stdout);
                nofvessel--;
                tables[i].cap = robot[j].cap;
                tables[i].state = 1;
            }
            pthread_mutex_unlock(&tables[i].tablelock);
        }
    }
    return;
}
void *tabls(void *a)
{
    sleep(10);
    struct timespec ts;
    clock_gettime(CLOCK_MONOTONIC, &ts);
```

This Snippet is used to assign table to vessel and change state of the TABLE.