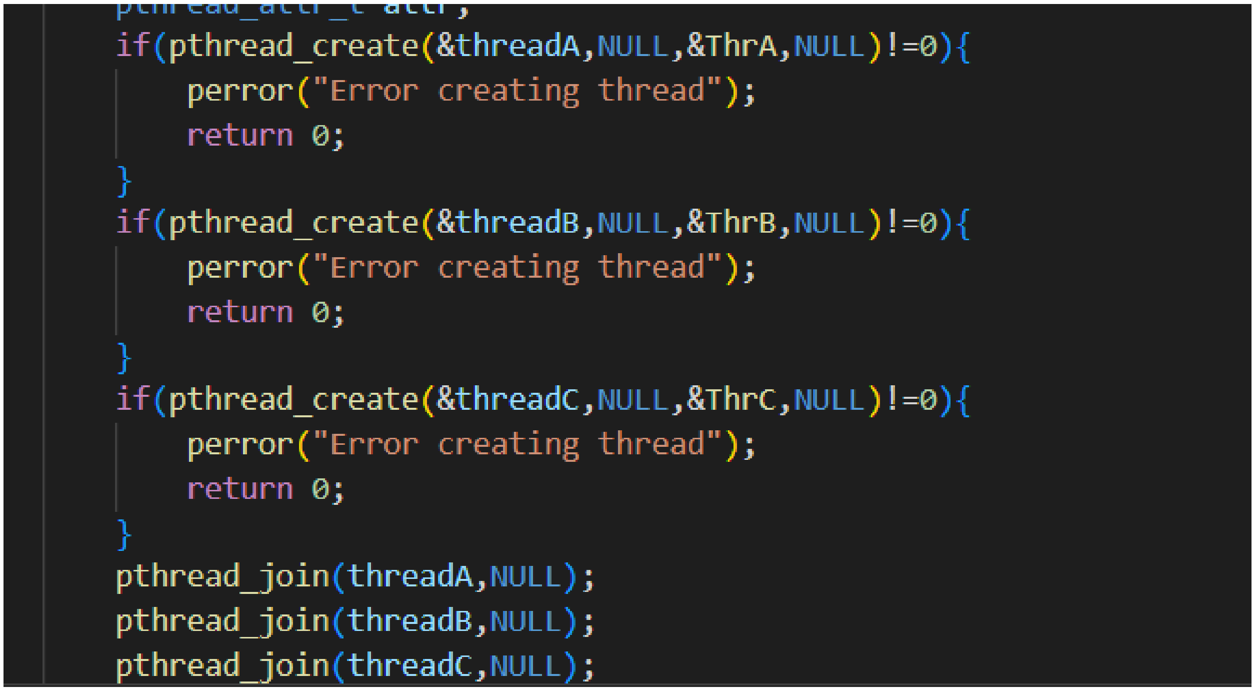
Assignment2

Question 1.1

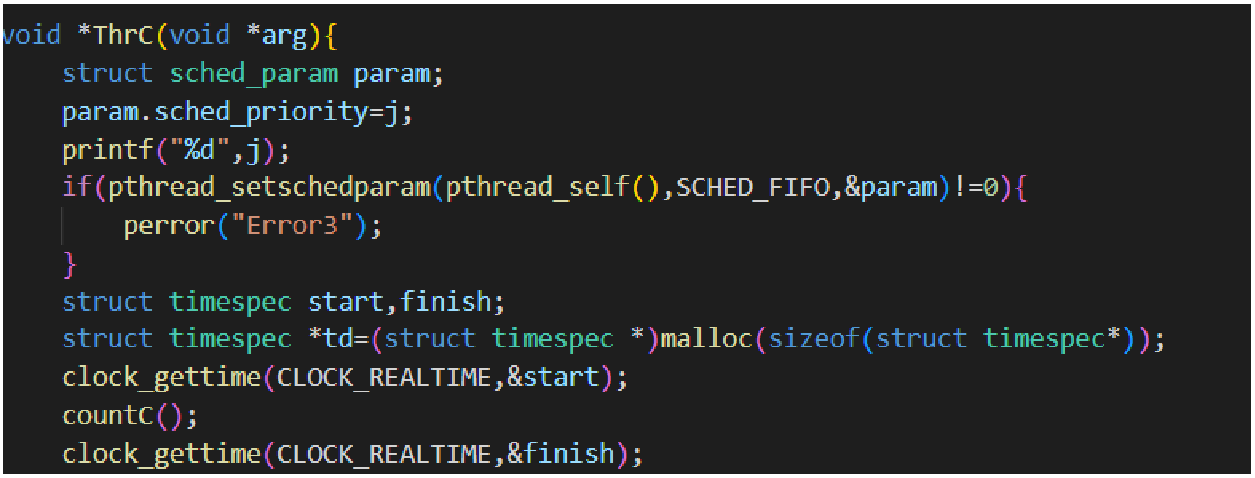
Three threads are created in this C program using pthread_create. These threads are used to call functions ThrA,ThrB,ThrC.



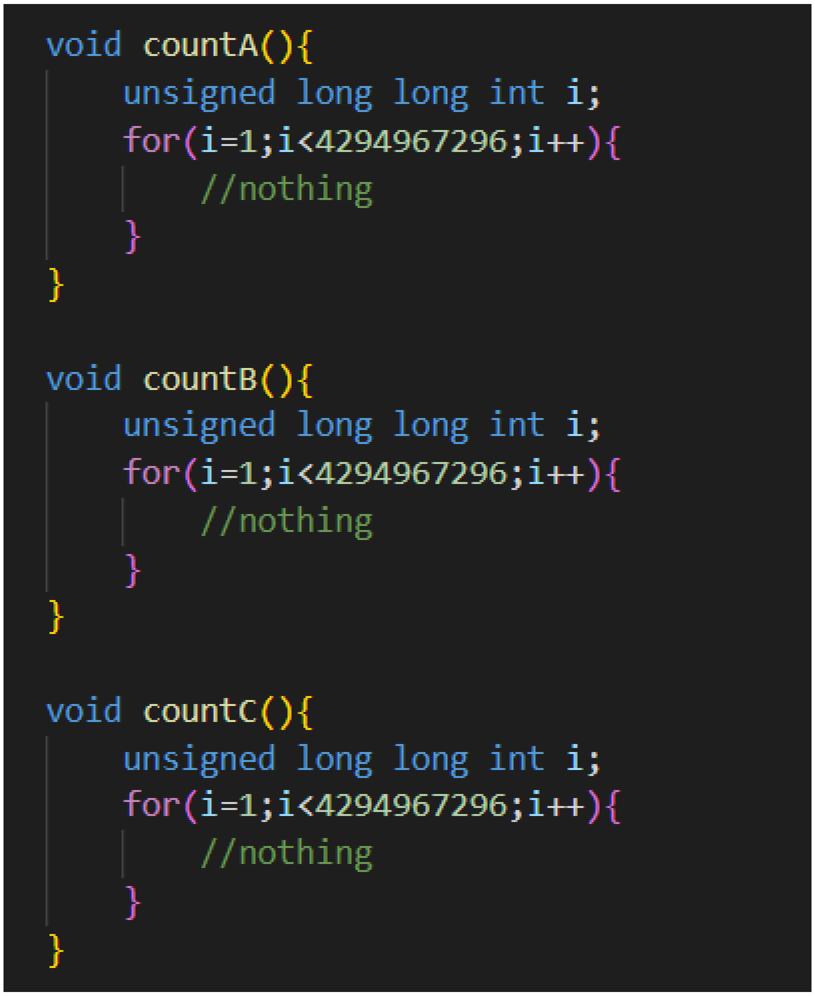
```
    pthread_attr_t attr;
    if(pthread_create(&threadA,NULL,&ThrA,NULL)!=0){
        perror("Error creating thread");
        return 0;
    }
    if(pthread_create(&threadB,NULL,&ThrB,NULL)!=0){
        perror("Error creating thread");
        return 0;
    }
    if(pthread_create(&threadC,NULL,&ThrC,NULL)!=0){
        perror("Error creating thread");
        return 0;
    }
    pthread_join(threadA,NULL);
    pthread_join(threadB,NULL);
    pthread_join(threadC,NULL);
```

These functions are used to set the priority and policy which is fixed at 0 from SCHED_OTHER but changes from 1 to 46 for SCHED_FIFO and SCHED_RR.

ThreadA-SCHED_OTHER
ThreadB-SCHED_RR
ThreadC-SCHED_FIFO
These functions are used to call countA,countB,countC which iterate from 1 to 2 raised to power 32 and calculate time taken.

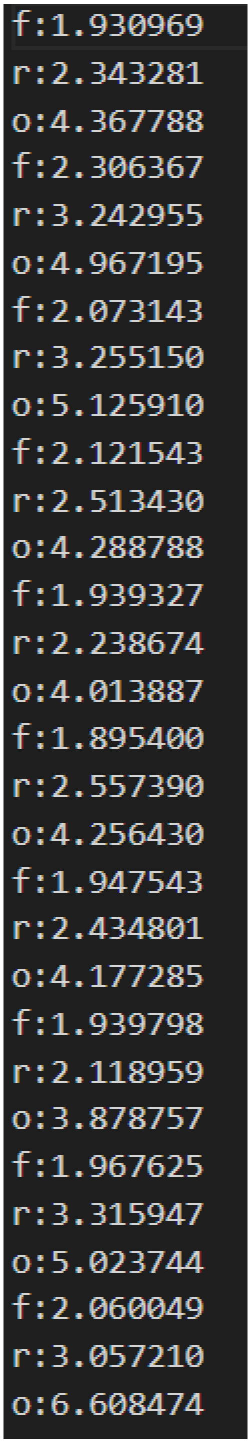


```c
void *ThrC(void *arg){
    struct sched_param param;
    param.sched_priority=j;
    printf("%d",j);
    if(pthread_setschedparam(pthread_self(),SCHED_FIFO,&param)!=0){
        perror("Error3");
    }
    struct timespec start,finish;
    struct timespec *td=(struct timespec *)malloc(sizeof(struct timespec*));
    clock_gettime(CLOCK_REALTIME,&start);
    countC();
    clock_gettime(CLOCK_REALTIME,&finish);
```

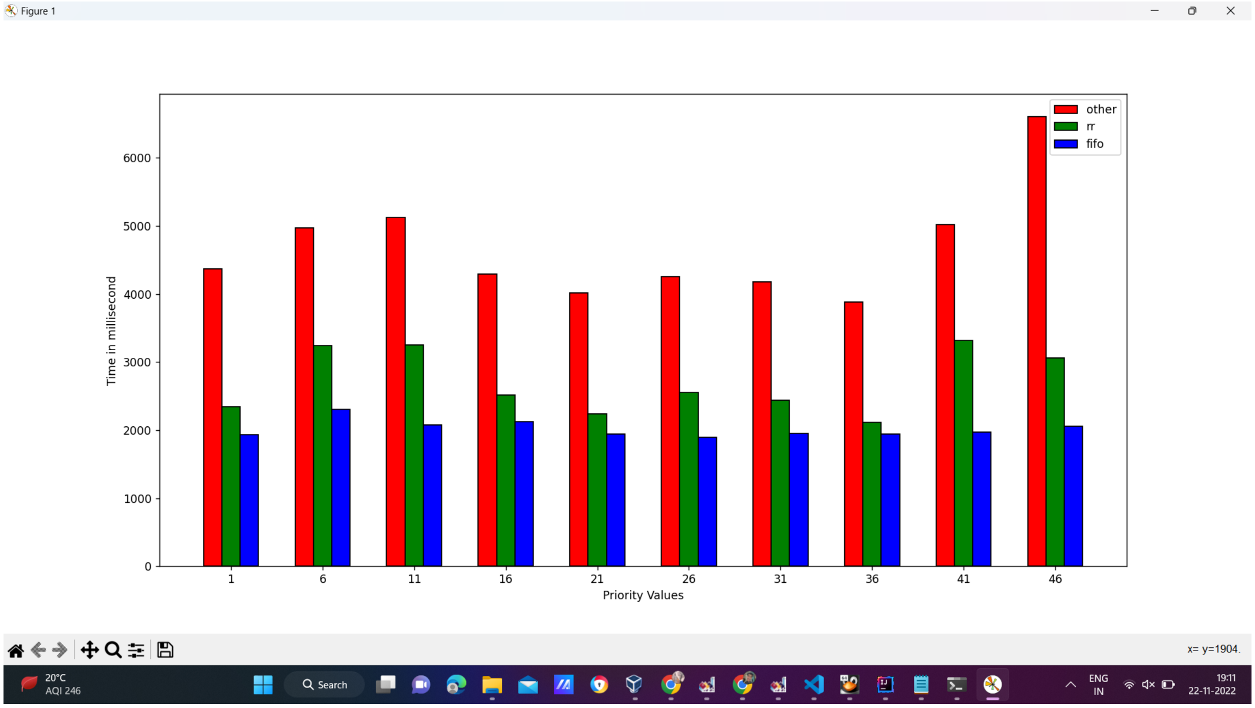


```c
void countA(){
    unsigned long long int i;
    for(i=1;i<4294967296;i++){
        //nothing
    }
}

void countB(){
    unsigned long long int i;
    for(i=1;i<4294967296;i++){
        //nothing
    }
}

void countC(){
    unsigned long long int i;
    for(i=1;i<4294967296;i++){
        //nothing
    }
}
```

The results are highly non deterministic. Fifo has a time of around 1.9-2 seconds. RR is slightly more at 2.3-2.5 seconds. Other is much higher at around 4-5 seconds. Such results are more or less reproducible for every iteration.



```
f:1.930969
r:2.343281
o:4.367788
f:2.306367
r:3.242955
o:4.967195
f:2.073143
r:3.255150
o:5.125910
f:2.121543
r:2.513430
o:4.288788
f:1.939327
r:2.238674
o:4.013887
f:1.895400
r:2.557390
o:4.256430
f:1.947543
r:2.434801
o:4.177285
f:1.939798
r:2.118959
o:3.878757
f:1.967625
r:3.315947
o:5.023744
f:2.060049
r:3.057210
o:6.608474
```



This is because SCHED_FIFO and SCHED_RR are real time scheduling policies which pre empt every other task. Fifo does not have a time slice so it is the fastest. RR has a time slice so it is slightly slower. Other on the hand is the default scheduling policy that schedules processes for a time slice depending on other processes.