SIMULATION BASED ASSIGNMENT

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

NAME OF THE STUDENT: Utkarsh Somvanshi

Mail ID: utkarshsomvanshi@gmail.com

Reg.no: 11710436

Roll No: B48

Section: K17UK

Link: https://github.com/utkarshsomvanshi/OS-Project

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Code :12

Description:

1) CPU Scheduling | Longest Remaining Time First (LRTF) algorithm
   We have given some process with arrival time and Burst Time and we have to find the completion time (CT), Turn Around Time(TAT), Average Turn Around Time (Avg TAT), Waiting Time(WT), Average Waiting Time (AWT) for the given processes.

2) First, sort the processes in increasing order of their Arrival Time.Choose the process having least arrival time but with most Burst Time. Then process it for 1 unit. Check if any other process arrives upto that time of execution or not. Repeat the above both steps until execute all the processes.

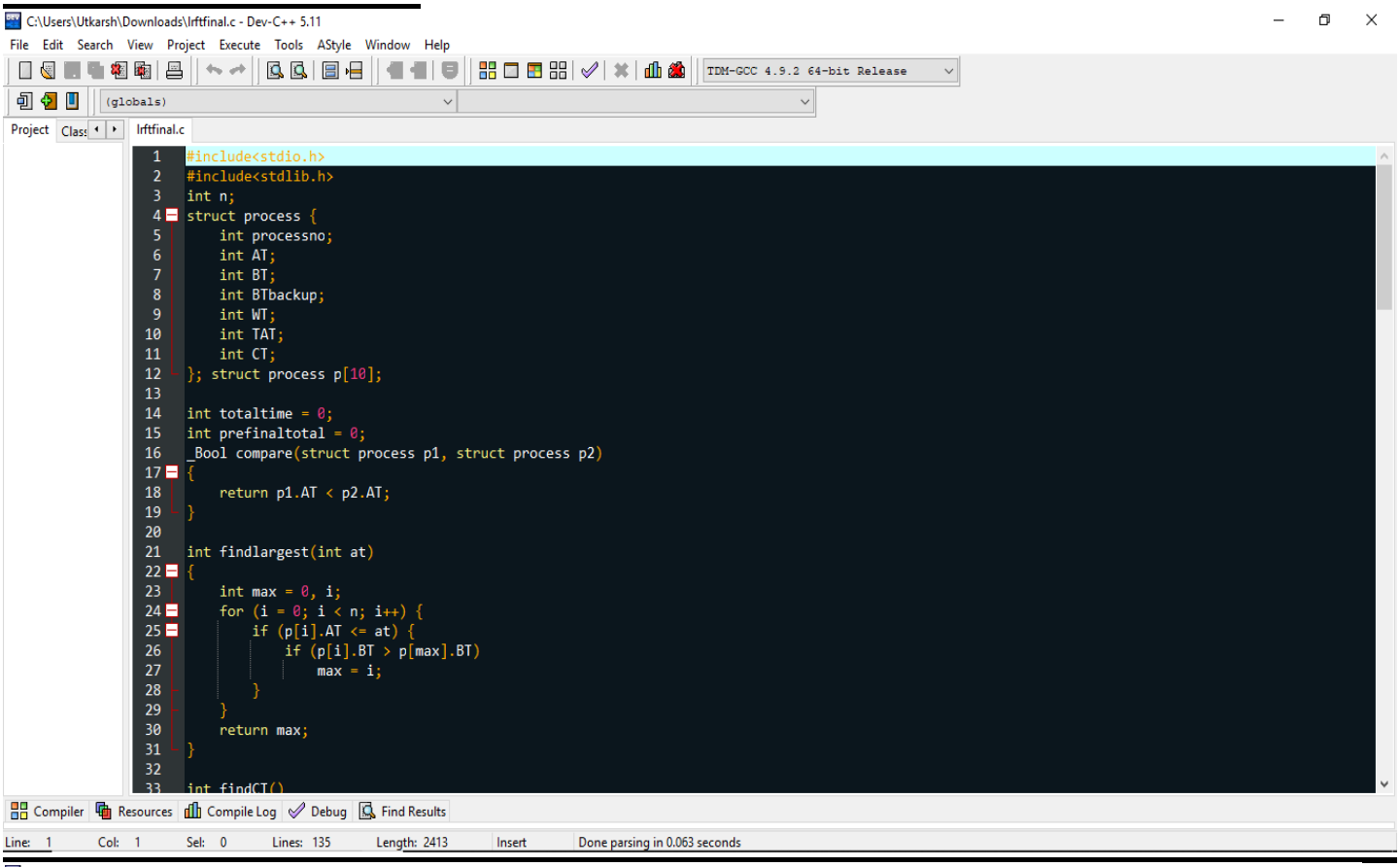
3) Turn Around time (TAT)
   =(Complition time) – (Arrival time)
   Also , Waiting time (WT)
   .=(Turn Around time) - (burst time).

Algorithm –

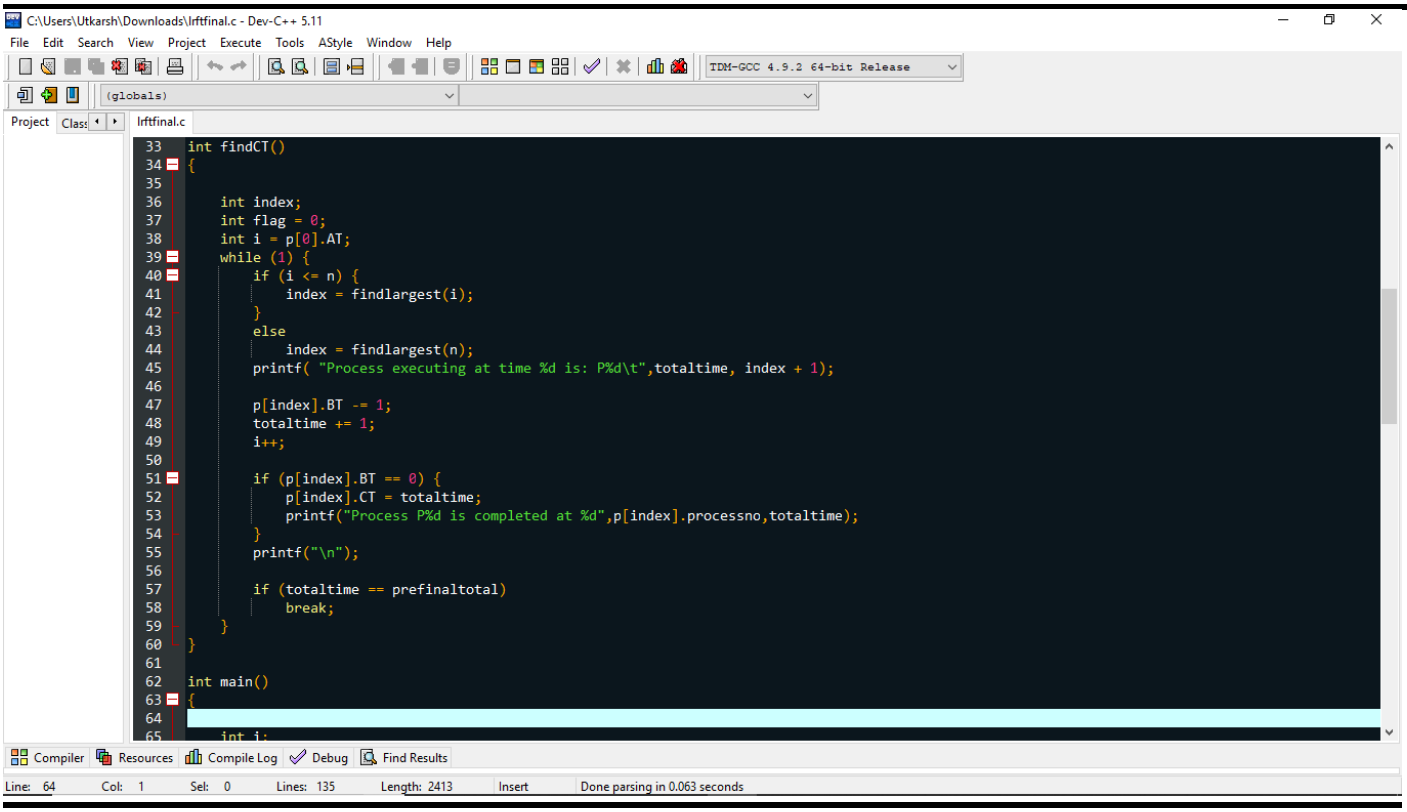
1 :- Create a structure of process containing all necessary fields like AT (Arrival Time), BT(Burst Time),
2 :-CT(Completion Time), TAT(Turn Around Time), WT(Waiting Time).
3 :- Sort according to the AT;
4 :- Find the process having Largest Burst Time and execute for each single unit. Increase the total time by 1 and reduce the Burst Time of that process with 1.
5 :- When any process have 0 BT left, then update the CT(Completion Time of that process CT will be Total Time at that time).
6:- After calculating the CT for each process, find TAT and WT.

(TAT =CT – AT)
(WT = TAT - BT)

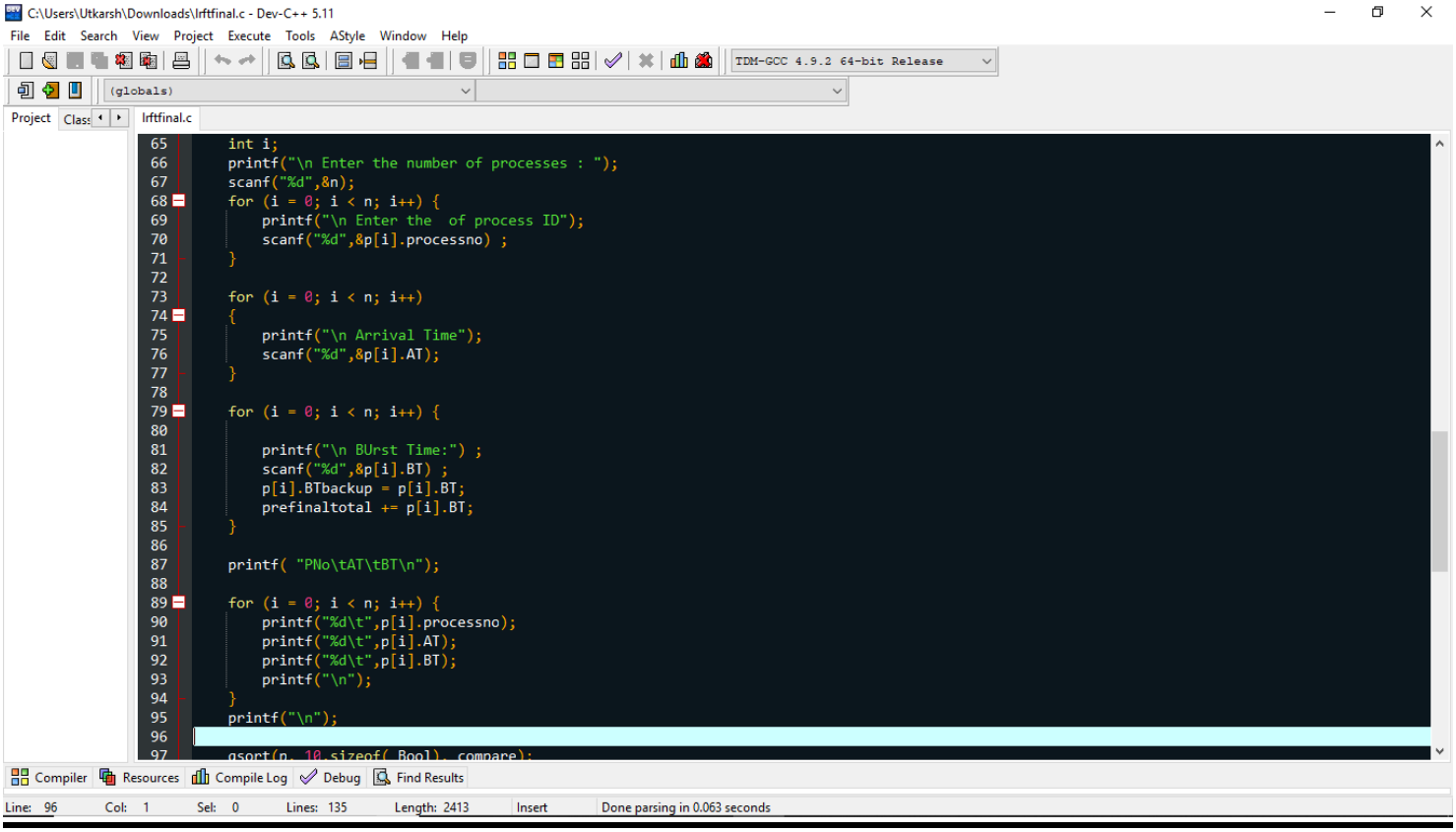
CODE SNIPPET :

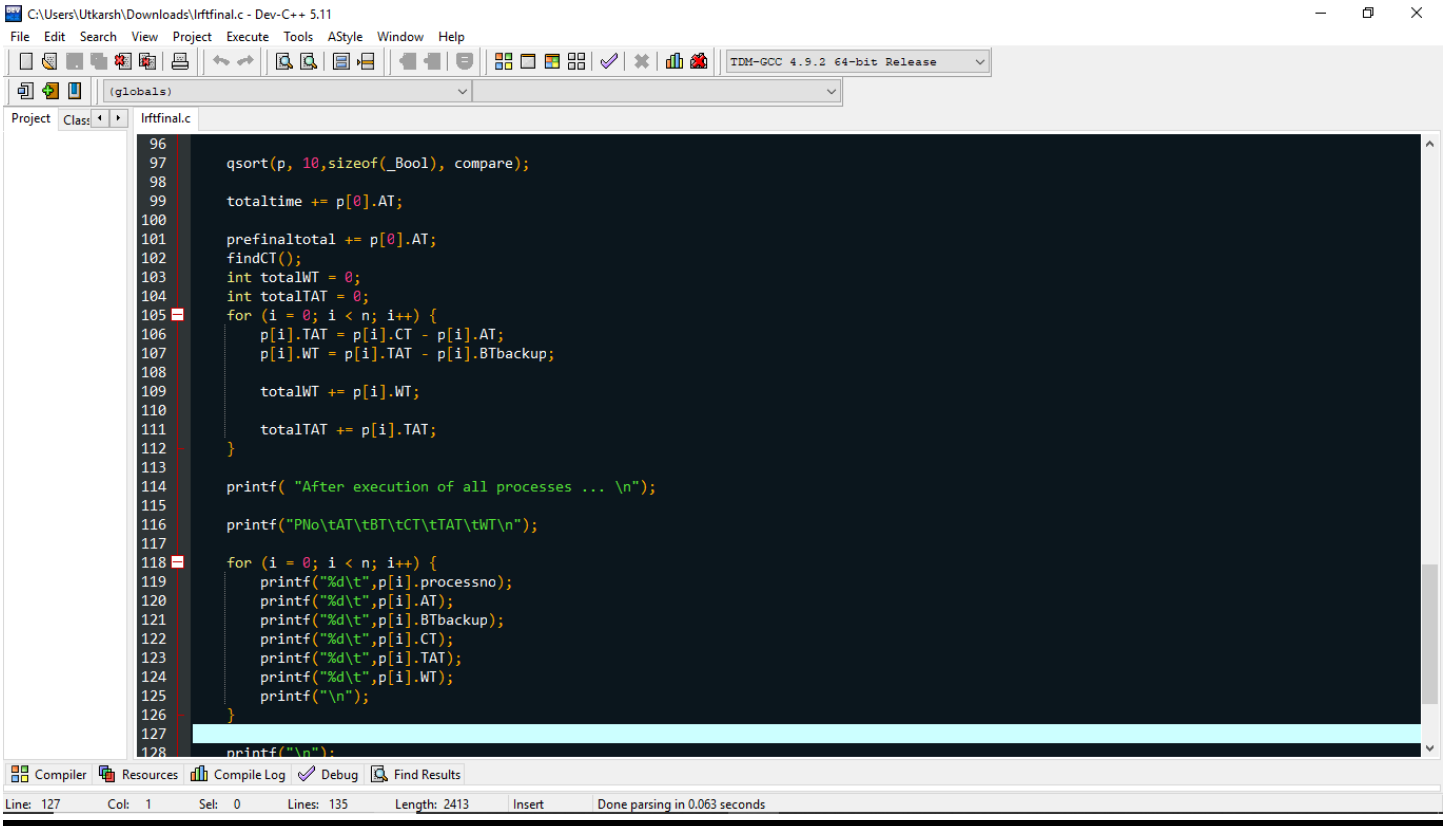


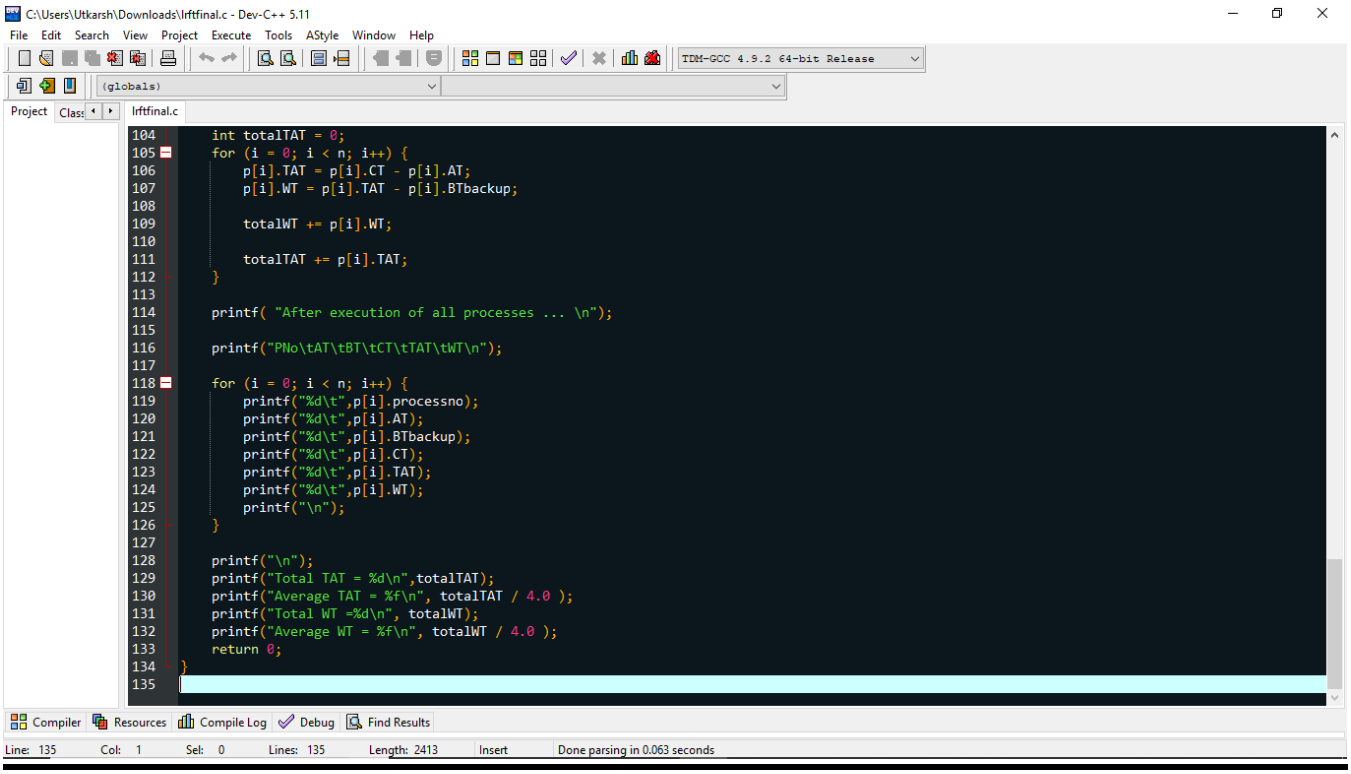

```c
#include<stdio.h>
#include<stdlib.h>
int n;
struct process {
    int processno;
    int AT;
    int BT;
    int BTbackup;
    int WT;
    int TAT;
    int CT;
}; struct process p[10];

int totaltime = 0;
int prefinaltotal = 0;
_Bool compare(struct process p1, struct process p2)
{
    return p1.AT < p2.AT;
}

int findlargest(int at)
{
    int max = 0, i;
    for (i = 0; i < n; i++) {
        if (p[i].AT <= at) {
            if (p[i].BT > p[max].BT)
                max = i;
        }
    }
    return max;
}

int findCT()
```



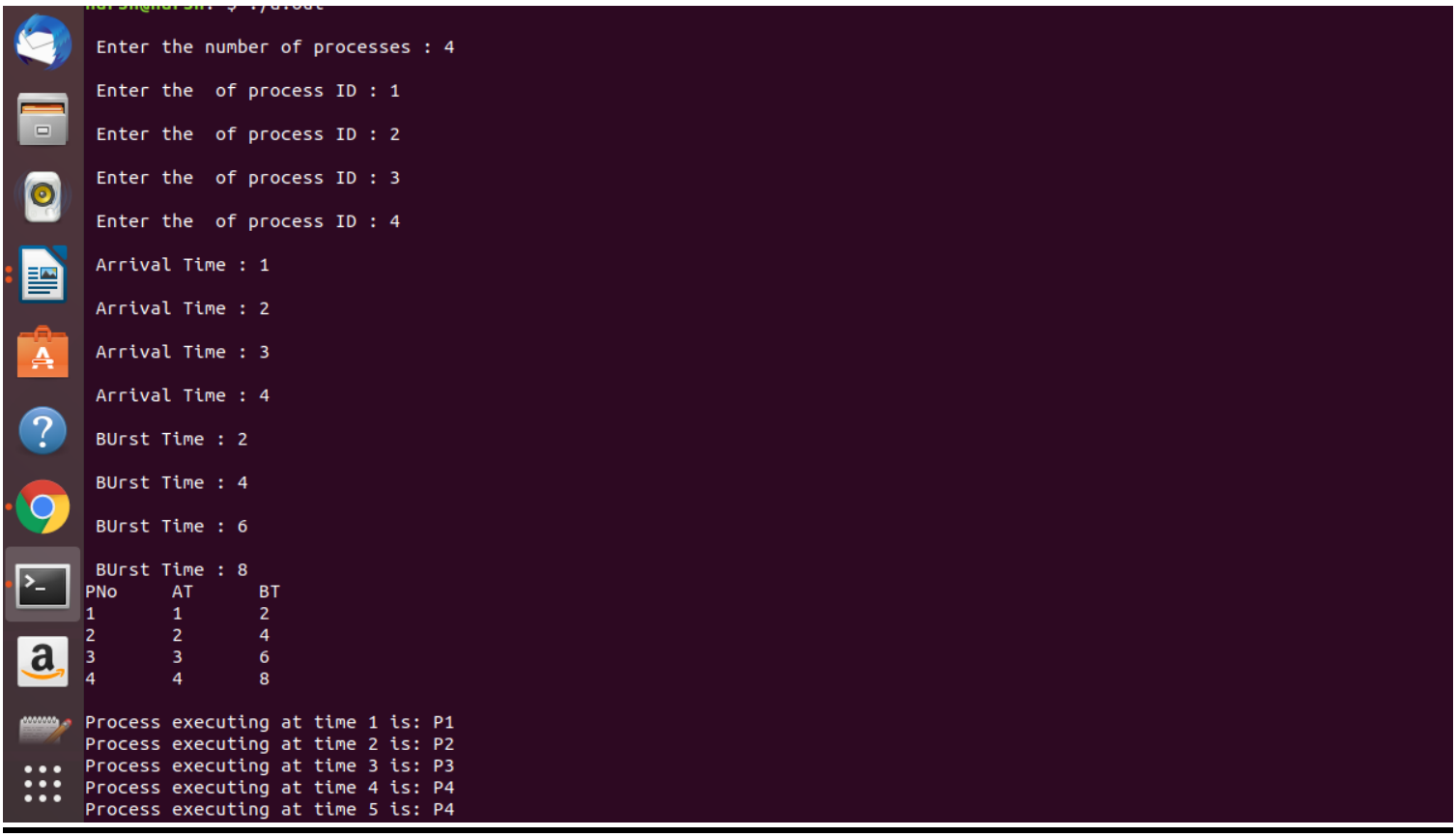

```c
int findCT()
{

    int index;
    int flag = 0;
    int i = p[0].AT;
    while (1) {
        if (i <= n) {
            index = findlargest(i);
        }
        else
            index = findlargest(n);
        printf( "Process executing at time %d is: P%d\t",totaltime, index + 1);

        p[index].BT -= 1;
        totaltime += 1;
        i++;

        if (p[index].BT == 0) {
            p[index].CT = totaltime;
            printf("Process P%d is completed at %d",p[index].processno,totaltime);
        }
        printf("\n");

        if (totaltime == prefinaltotal)
            break;
    }
}

int main()
{

    int i;
```

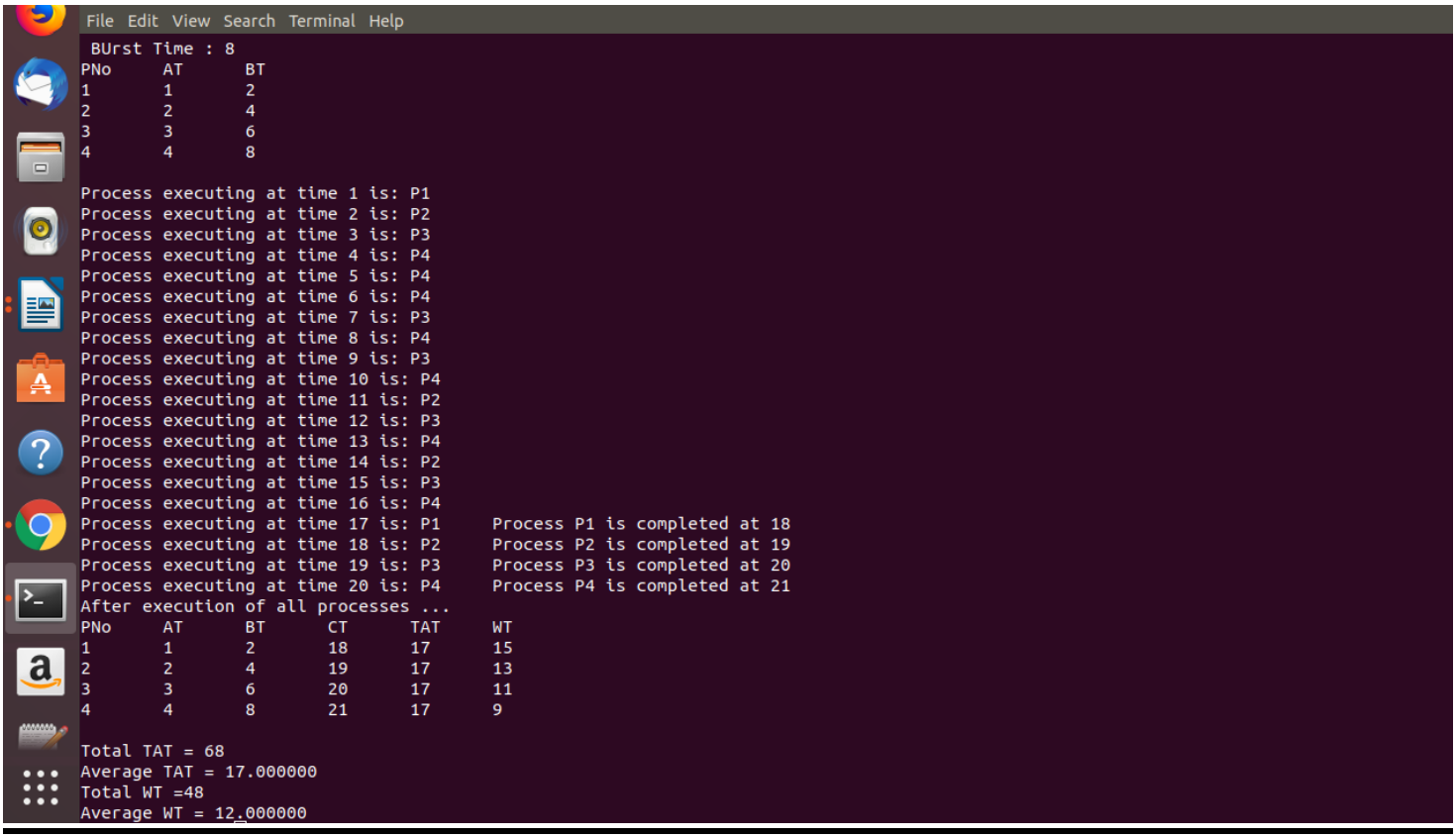

C:\Users\Utkarsh\Downloads\lrftfinal.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Class
lrftfinal.c
int i;
printf("\n Enter the number of processes : ");
scanf("%d",&n);
for (i = 0; i < n; i++) {
printf("\n Enter the of process ID");
scanf("%d",&p[i].processno) ;
}
for (i = 0; i < n; i++)
{
printf("\n Arrival Time");
scanf("%d",&p[i].AT);
}
for (i = 0; i < n; i++) {
printf("\n BUrst Time:") ;
scanf("%d",&p[i].BT) ;
p[i].BTbackup = p[i].BT;
prefinaltotal += p[i].BT;
}
printf("PNo\tAT\tBT\n");
for (i = 0; i < n; i++) {
printf("%d\t",p[i].processno);
printf("%d\t",p[i].AT);
printf("%d\t",p[i].BT);
printf("\n");
}
printf("\n");
Compiler Resources Compile Log Debug Find Results
Line: 96 Col: 1 Sel: 0 Lines: 135 Length: 2413 Insert Done parsing in 0.063 seconds


C:\Users\Utkarsh\Downloads\lrftfinal.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Class
lrftfinal.c
qsort(p, 10,sizeof(_Bool), compare);
totaltime += p[0].AT;
prefinaltotal += p[0].AT;
findCT();
int totalWT = 0;
int totalTAT = 0;
for (i = 0; i < n; i++) {
p[i].TAT = p[i].CT - p[i].AT;
p[i].WT = p[i].TAT - p[i].BTbackup;
totalWT += p[i].WT;
totalTAT += p[i].TAT;
}
printf("After execution of all processes ... \n");
printf("PNo\tAT\tBT\tCT\tTAT\tWT\n");
for (i = 0; i < n; i++) {
printf("%d\t",p[i].processno);
printf("%d\t",p[i].AT);
printf("%d\t",p[i].BTbackup);
printf("%d\t",p[i].CT);
printf("%d\t",p[i].TAT);
printf("%d\t",p[i].WT);
printf("\n");
}
Compiler Resources Compile Log Debug Find Results
Line: 127 Col: 1 Sel: 0 Lines: 135 Length: 2413 Insert Done parsing in 0.063 seconds


C:\Users\Utkarsh\Downloads\lrftfinal.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Class
lrftfinal.c
int totalTAT = 0;
for (i = 0; i < n; i++) {
p[i].TAT = p[i].CT - p[i].AT;
p[i].WT = p[i].TAT - p[i].BTbackup;
totalWT += p[i].WT;
totalTAT += p[i].TAT;
}
printf("After execution of all processes ... \n");
printf("PNo\tAT\tBT\tCT\tTAT\tWT\n");
for (i = 0; i < n; i++) {
printf("%d\t",p[i].processno);
printf("%d\t",p[i].AT);
printf("%d\t",p[i].BTbackup);
printf("%d\t",p[i].CT);
printf("%d\t",p[i].TAT);
printf("%d\t",p[i].WT);
printf("\n");
}
printf("\n");
printf("Total TAT = %d\n",totalTAT);
printf("Average TAT = %f\n", totalTAT / 4.0);
printf("Total WT =%d\n", totalWT);
printf("Average WT = %f\n", totalWT / 4.0);
return 0;
}
Compiler Resources Compile Log Debug Find Results
Line: 135 Col: 1 Sel: 0 Lines: 135 Length: 2413 Insert Done parsing in 0.063 seconds

Test cases -;


Enter the number of processes : 4
Enter the of process ID : 1
Enter the of process ID : 2
Enter the of process ID : 3
Enter the of process ID : 4
Arrival Time : 1
Arrival Time : 2
Arrival Time : 3
Arrival Time : 4
BUrst Time : 2
BUrst Time : 4
BUrst Time : 6
BUrst Time : 8
PNo AT BT
1 1 2
2 2 4
3 3 6
4 4 8
Process executing at time 1 is: P1
Process executing at time 2 is: P2
Process executing at time 3 is: P3
Process executing at time 4 is: P4
Process executing at time 5 is: P4



In the following project the test cases provided aboves shows the result obtained by lrtf (longest remaining time first algorthims and the data inputed in it are inr following manner and thus inputs are given as follows:-

Id is inputed:- 4
And in the question is (a,b,c)
2132
2102
2453
And the time taken in mess 2,4,8

Github link: https://github.com/utkarshsomvanshi/OS-Project
