

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>



Transforming Education Transforming India

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Code :12

Description:

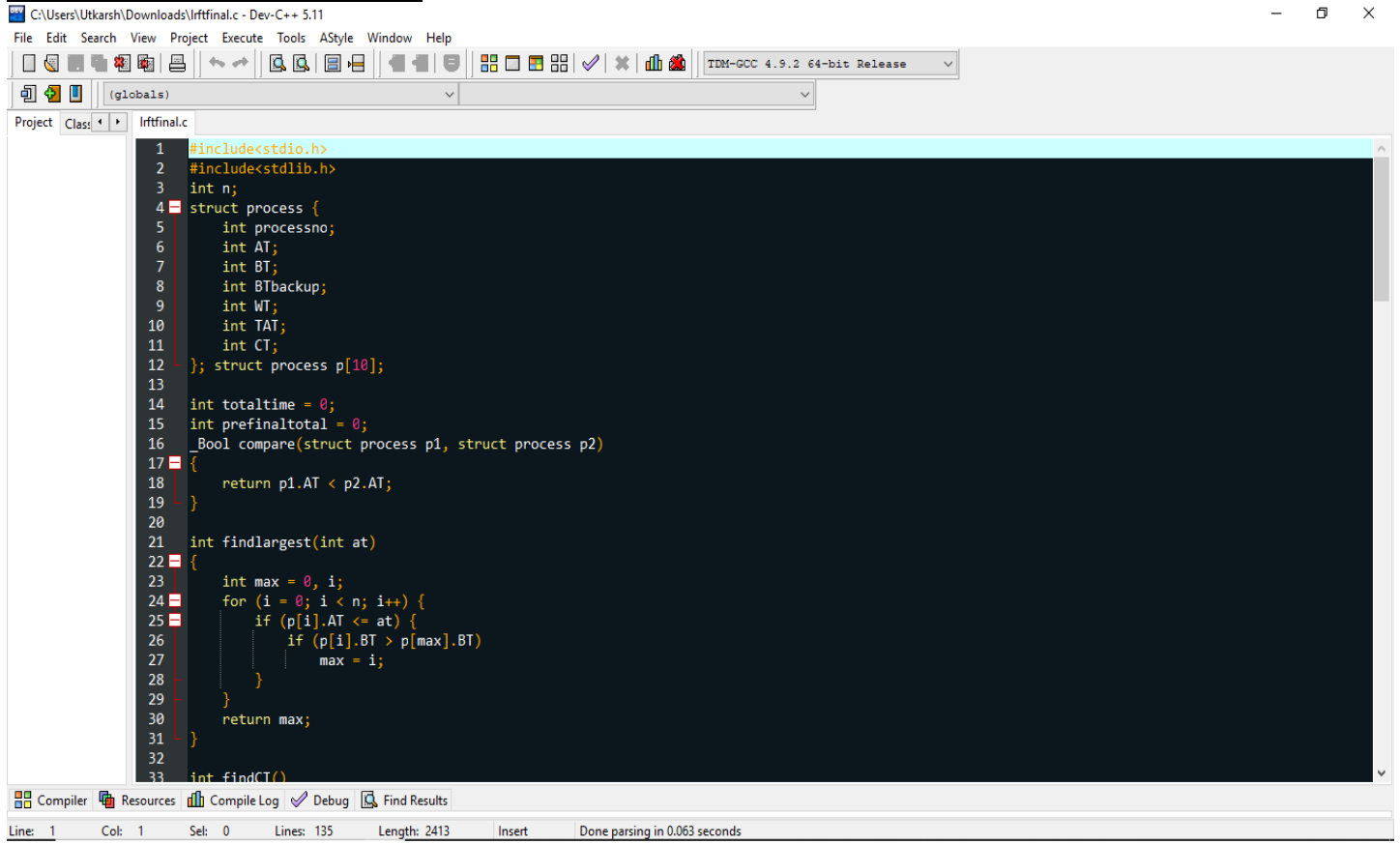
- 1) CPU Scheduling | Longest Remaining Time First (LRTF) algorithm
We have given some process with arrival time and Burst Time and we have to find the completion time (CT), Turn Around Time(TAT), Average Turn Around Time (Avg TAT), Waiting Time(WT), Average Waiting Time (AWT) for the given processes.
- 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)
 $\text{TAT} = (\text{Completion time}) - (\text{Arrival time})$
Also , Waiting time (WT)
 $\text{WT} = (\text{Turn Around time}) - (\text{burst time})$.

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.

$$\text{TAT} = \text{CT} - \text{AT}$$
$$\text{WT} = \text{TAT} - \text{BT}$$

CODE SNIPPET :



C:\Users\Utkarsh\Downloads\lrfinal.c - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

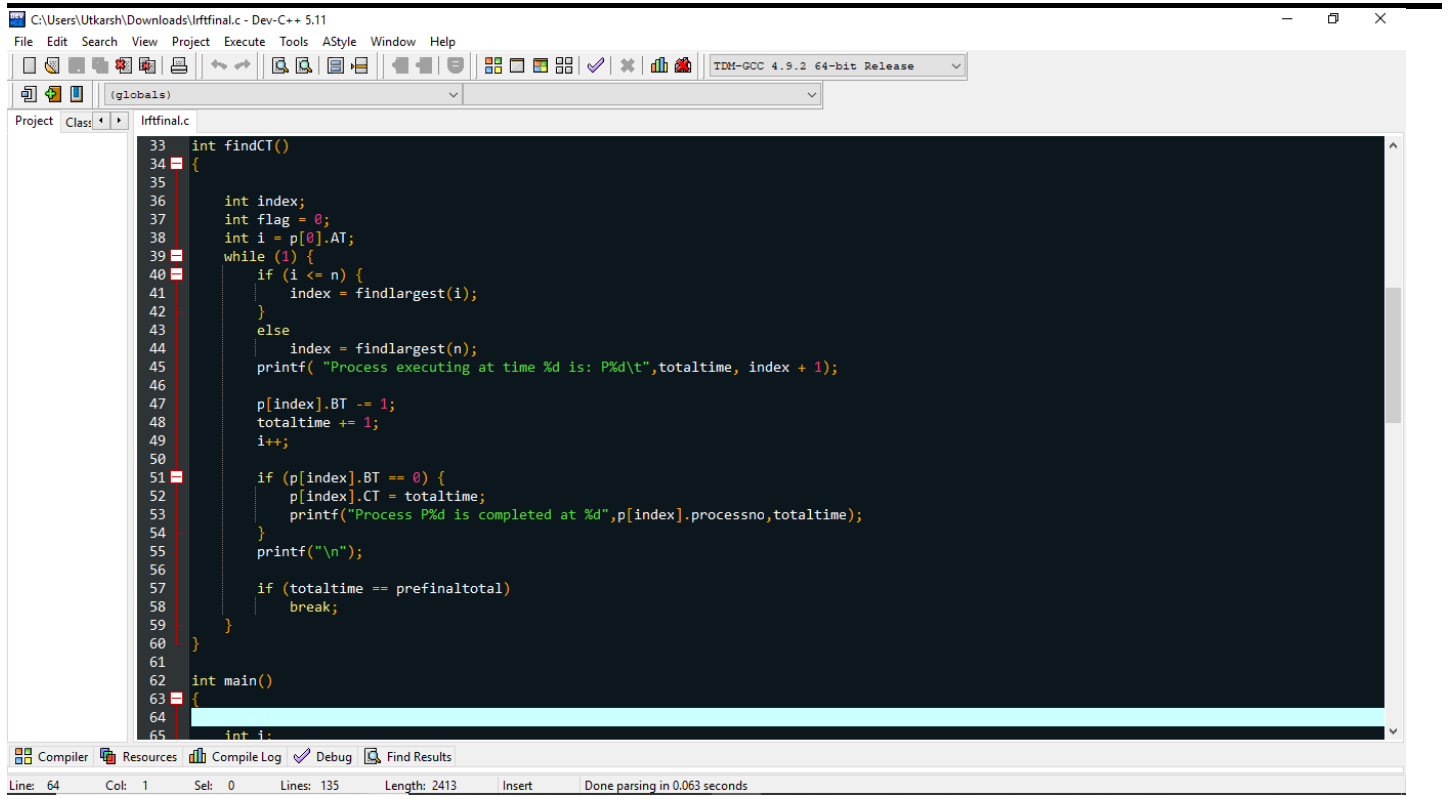
(globals)

Project Class lrfinal.c

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 int n;
4 struct process {
5     int processno;
6     int AT;
7     int BT;
8     int BTbackup;
9     int WT;
10    int TAT;
11    int CT;
12 }; struct process p[10];
13
14 int totaltime = 0;
15 int prefinaltotal = 0;
16 _Bool compare(struct process p1, struct process p2)
17 {
18     return p1.AT < p2.AT;
19 }
20
21 int findlargest(int at)
22 {
23     int max = 0, i;
24     for (i = 0; i < n; i++) {
25         if (p[i].AT <= at) {
26             if (p[i].BT > p[max].BT)
27                 max = i;
28         }
29     }
30     return max;
31 }
32
33 int findCT()
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 135 Length: 2413 Insert Done parsing in 0.063 seconds



C:\Users\Utkarsh\Downloads\lrfinal.c - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Class lrfinal.c

```
33 int findCT()
34 {
35
36     int index;
37     int flag = 0;
38     int i = p[0].AT;
39     while (1) {
40         if (i <= n) {
41             index = findlargest(i);
42         }
43         else
44             index = findlargest(n);
45         printf("Process executing at time %d is: P%d\t", totaltime, index + 1);
46
47         p[index].BT -= 1;
48         totaltime += 1;
49         i++;
50
51         if (p[index].BT == 0) {
52             p[index].CT = totaltime;
53             printf("Process P%d is completed at %d", p[index].processno, totaltime);
54         }
55         printf("\n");
56
57         if (totaltime == prefinaltotal)
58             break;
59     }
60 }
61
62 int main()
63 {
64
65     int i;
```

Compiler Resources Compile Log Debug Find Results

Line: 64 Col: 1 Sel: 0 Lines: 135 Length: 2413 Insert Done parsing in 0.063 seconds

C:\Users\Utkarsh\Downloads\lrfinal.c - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Class lrfinal.c

```
65 int i;
66 printf("\n Enter the number of processes : ");
67 scanf("%d",&n);
68 for (i = 0; i < n; i++) {
69     printf("\n Enter the of process ID");
70     scanf("%d",&p[i].processno);
71 }
72
73 for (i = 0; i < n; i++)
74 {
75     printf("\n Arrival Time");
76     scanf("%d",&p[i].AT);
77 }
78
79 for (i = 0; i < n; i++) {
80
81     printf("\n BUnst Time:") ;
82     scanf("%d",&p[i].BT) ;
83     p[i].BTbackup = p[i].BT;
84     prefinaltotal += p[i].BT;
85 }
86
87 printf( "PNo\tAT\tBT\n");
88
89 for (i = 0; i < n; i++) {
90     printf("%d\t",p[i].processno);
91     printf("%d\t",p[i].AT);
92     printf("%d\t",p[i].BT);
93     printf("\n");
94 }
95 printf("\n");
96
97 qsort(p, 10, sizeof(_Bool), compare);
```

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\lrfinal.c - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Class lrfinal.c

```
96
97 qsort(p, 10, sizeof(_Bool), compare);
98
99 totaltime += p[0].AT;
100
101 prefinaltotal += p[0].AT;
102 findCT();
103 int totalWT = 0;
104 int totalTAT = 0;
105 for (i = 0; i < n; i++) {
106     p[i].TAT = p[i].CT - p[i].AT;
107     p[i].WT = p[i].TAT - p[i].BTbackup;
108
109     totalWT += p[i].WT;
110
111     totalTAT += p[i].TAT;
112 }
113
114 printf( "After execution of all processes ... \n");
115
116 printf("PNo\tAT\tBT\tCT\tTAT\tWT\n");
117
118 for (i = 0; i < n; i++) {
119     printf("%d\t",p[i].processno);
120     printf("%d\t",p[i].AT);
121     printf("%d\t",p[i].BTbackup);
122     printf("%d\t",p[i].CT);
123     printf("%d\t",p[i].TAT);
124     printf("%d\t",p[i].WT);
125     printf("\n");
126 }
127
128 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\lrfinal.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Class lrfinal.c
104 int totalTAT = 0;
105 for (i = 0; i < n; i++) {
106     p[i].TAT = p[i].CT - p[i].AT;
107     p[i].WT = p[i].TAT - p[i].BTbackup;
108
109     totalWT += p[i].WT;
110
111     totalTAT += p[i].TAT;
112 }
113
114 printf( "After execution of all processes ... \n");
115
116 printf("PNo\tAT\tBT\tCT\tTAT\tWT\n");
117
118 for (i = 0; i < n; i++) {
119     printf("%d\t",p[i].processno);
120     printf("%d\t",p[i].AT);
121     printf("%d\t",p[i].BTbackup);
122     printf("%d\t",p[i].CT);
123     printf("%d\t",p[i].TAT);
124     printf("%d\t",p[i].WT);
125     printf("\n");
126 }
127
128 printf("\n");
129 printf("Total TAT = %d\n",totalTAT);
130 printf("Average TAT = %f\n", totalTAT / 4.0 );
131 printf("Total WT =%d\n", totalWT);
132 printf("Average WT = %f\n", totalWT / 4.0 );
133 return 0;
134 }
135
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 -;

```
not signed in - 1/8/2022
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
```

```
File Edit View Search Terminal Help
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
Process executing at time 6 is: P4
Process executing at time 7 is: P3
Process executing at time 8 is: P4
Process executing at time 9 is: P3
Process executing at time 10 is: P4
Process executing at time 11 is: P2
Process executing at time 12 is: P3
Process executing at time 13 is: P4
Process executing at time 14 is: P2
Process executing at time 15 is: P3
Process executing at time 16 is: P4
Process executing at time 17 is: P1
Process executing at time 18 is: P2
Process executing at time 19 is: P3
Process executing at time 20 is: P4
After execution of all processes ...
PNo  AT  BT  CT  TAT  WT
1    1   2   18   17   15
2    2   4   19   17   13
3    3   6   20   17   11
4    4   8   21   17    9

Total TAT = 68
Average TAT = 17.000000
Total WT = 48
Average WT = 12.000000

Process P1 is completed at 18
Process P2 is completed at 19
Process P3 is completed at 20
Process P4 is completed at 21
```

In the following project the test cases provided above shows the result obtained by lrtf (longest remaining time first) algorithms and the data inputted in it are in the following manner and thus inputs are given as follows:-

Id is inputted:- 4

And in the question is (a,b,c)

2132

2102

2453

And the time taken in ms is 2,4,8

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