



Experiment No.10
Write a C Program to calculate AWT AND ATAT Using Round Robin algorithm

Student Name: Neha Sharma

Branch:20BIT-1

Semester: 3

Subject Name: OS LAB

UID:20BCS4576

Section/Group: A

Date of Performance:25/11/21

Subject Code: 21O-20CSP-232

1. Aim/Overview of the practical:

Write a C Program to calculate AWT AND ATAT Using Round Robin algorithm

2. Task to be done:

Write a C Program to calculate AWT AND ATAT Using Round Robin algorithm

3. Apparatus:

- Laptop/PC
- Good internet connection
- C language IDE
- UNIX system

4. Steps for experiment/practical:

Step 1: Organize all processes according to their arrival time in the ready queue. The queue structure of the ready queue is based on the FIFO structure to execute all CPU processes.

Step 2: Now, we push the first process from the ready queue to execute its task for a fixed time, allocated by each process that arrives in the queue.

Step 3: If the process cannot complete their task within defined time interval or slots because it is stopped by another process that pushes from the ready queue to execute their task due to arrival time of the next process is reached. Therefore, CPU saved the previous state of the process, which helps to resume from the point where it is interrupted. (If the burst time of the process is left, push the process end of the ready queue).

Step 4: Similarly, the scheduler selects another process from the ready queue to execute its tasks. When a process finishes its task within time slots, the process will not go for further execution because the process's burst time is finished.

Step 5: Similarly, we repeat all the steps to execute the process until the work has finished

PROGRAM:-

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int i, NOP, sum=0,count=0, y, quant, wt=0, tat=0, at[10], bt[10],
    temp[10];
    float avg_wt, avg_tat;
    printf(" Total number of process in the system: ");
    scanf("%d", &NOP);
    y = NOP;

    for(i=0; i<NOP; i++)
    {
        printf("\n Enter the Arrival and Burst time of the Process[%d]\n", i+1);
        printf(" Arrival time is: \t");
        scanf("%d", &at[i]);
        printf(" \nBurst time is: \t");
        scanf("%d", &bt[i]);
        temp[i] = bt[i];
    }
    printf("Enter the Time Quantum for the process: \t");
    scanf("%d", &quant);
    printf("\n Process No \t\t Burst Time \t\t TAT \t\t Waiting Time ");
    for(sum=0, i = 0; y!=0; )
    {
        if(temp[i] <= quant && temp[i] > 0) {
```



```
sum = sum + temp[i];
temp[i] = 0;
count=1;
}
else if(temp[i] > 0)
{
temp[i] = temp[i] - quant;
sum = sum + quant;
}
if(temp[i]==0 && count==1)
{
y--;
printf("\nProcess No[%d] \t\t %d\t\t\t\t %d\t\t\t\t %d", i+1, bt[i],
sum-at[i], sum-at[i]-bt[i]);
wt = wt+sum-at[i]-bt[i];
tat = tat+sum-at[i];
count =0;
}
if(i==NOP-1)
{
i=0;
}
else if(at[i+1]<=sum)
{
i++;
}
else
{
```



DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.



```
i=0;  
}  
}  
avg_wt = wt * 1.0/NOP;  
avg_tat = tat * 1.0/NOP;  
printf("\n Average Turn Around Time: \t%f", avg_wt);  
printf("\n Average Waiting Time: \t%f", avg_tat);  
getch();  
}
```

OUTPUT SCREEN



DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

NAAC
GRADE **A+**
ACCREDITED UNIVERSITY

GDB online Debugger | Compiler x

onlinegdb.com

Apps Google लॉग इन Gmail YouTube Maps Advanced Compute... View Assessment

Reading list

OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online

input

Total number of process in the system: 3
Enter the Arrival and Burst time of the Process[1]
Arrival time is: 23
Burst time is: 32
Enter the Arrival and Burst time of the Process[2]
Arrival time is: 12
Burst time is: 21
Enter the Arrival and Burst time of the Process[3]
Arrival time is: 45
Burst time is: 31
Enter the Time Quantum for the process: 6

Process No	Burst Time	TAT	Waiting Time
Process No[2]	21	39	18
Process No[1]	32	36	4
Process No[3]	31	39	8

Average Turn Around Time: 10.000000
Average Waiting Time: 38.000000
...Program finished with exit code 0
Press ENTER to exit console.

Type here to search

27

W

X

?

15°C

ENG

22:37

22-11-2021



Learning outcomes (What I have learnt):

1. Learn about how to use different Linux command
2. What is shell programming?
3. UNIX commands.
4. Shell script

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			