Lab Experiment Report No: 01

Implementation of FCFS

Process Scheduling Algorithm in C Language

1. Objective:

The objective of this lab experiment is to understand and implement the First-Come-First-Serve (FCFS) scheduling algorithm in the context of operating systems. FCFS is one of the simplest CPU scheduling algorithms where processes are executed based on their arrival time. The processes are executed in the order they arrive, without considering their burst times or any other factors.

2. Theory:

FCFS scheduling works by scheduling processes in the order they arrive in the ready queue. When a process arrives, it joins the end of the queue. The CPU then selects the process at the front of the queue for execution. Once a process starts execution, it continues until it completes its CPU burst. FCFS is non-preemptive, meaning once a process starts executing, it cannot be interrupted until it finishes its CPU burst.

3. Algorithm:

The algorithm is given by:

```
1- Input the processes along with their burst time (bt).
2- Find waiting time (wt) for all processes.
3-  s first process that comes need not to wait so
    waiting time for process 1 will be 0 i.e. wt[0] = 0.
4-  Find waiting time for all other processes i.e. for
     process i ->
       wt[i] = bt[i-1] + wt[i-1] .
5-  Find turnaround time = waiting_time + burst_time
    for all processes.
6-  Find average waiting time =
                total_waiting_time / no_of_processes.
```

7- Similarly, find average turnaround time =

total_turn_around_time / no_of_processes.

4. Source Code

The implementation of the FCFS algorithm involves creating a queue to hold processes and executing them in the order they arrive. The processes are executed one after another until all processes are completed.

Below is a sample code implementing the FCFS scheduling algorithm in C:

```
#include<stdio.h>
#include<conio.h>
int main()
{
   int  p[10],at[10],bt[10],ct[10],tat[10],wt[10],i,j,temp=0,n;
   float awt=0,atat=0;
   printf("enter no of proccess you want:");
   scanf("%d",&n);
   printf("enter %d process:",n);
   for(i=0;i<n;i++)
   {
   scanf("%d",&p[i]);
   }
   printf("enter %d arrival time:",n);
   for(i=0;i<n;i++)
   {
   scanf("%d",&at[i]);
   }
   printf("enter %d burst time:",n);
   for(i=0;i<n;i++)
   {
   scanf("%d",&bt[i]);
```

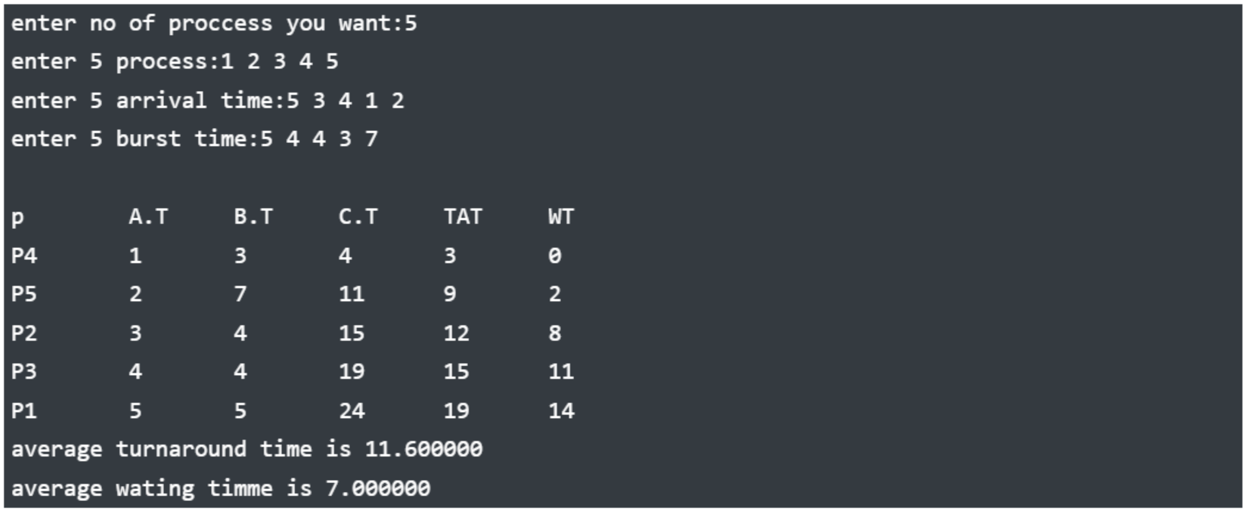
```
}
// sorting at,bt, and process according to at
for(i=0;i<n;i++)
{
 for(j=0;j<(n-i);j++)
{
  if(at[j]>at[j+1])
 {
    temp=p[j+1];
    p[j+1]=p[j];
    p[j]=temp;
    temp=at[j+1];
    at[j+1]=at[j];
    at[j]=temp;
    temp=bt[j+1];
    bt[j+1]=bt[j];
    bt[j]=temp;
  }
 }
}
/* calculating 1st ct */
ct[0]=at[0]+bt[0];
/* calculating 2 to n ct */
for(i=1;i<n;i++)
{
  //when proess is ideal in between i and i+1
  temp=0;
 if(ct[i-1]<at[i])
 {
    temp=at[i]-ct[i-1];
 }
```

```
     ct[i]=ct[i-1]+bt[i]+temp;
    }
    /* calculating tat and wt */
    printf("\np\t A.T\t B.T\t C.T\t TAT\t WT");
    for(i=0;i<n;i++)
    {
    tat[i]=ct[i]-at[i];
    wt[i]=tat[i]-bt[i];
    atat+=tat[i];
    awt+=wt[i];
    }
    atat=atat/n;
    awt=awt/n;
    for(i=0;i<n;i++)
    {
      printf("\nP%d\t %d\t %d\t %d \t %d \t %d",p[i],at[i],bt[i],ct[i],tat[i],wt[i]);
    }
    printf("\naverage turnaround time is %f",atat);


    printf("\naverage wating timme is %f",awt);
   getch();
    return 0;
}
```

5. Result and analysis:

After executing the FCFS algorithm, we obtained the following output:



```
enter no of proccess you want:5
enter 5 process:1 2 3 4 5
enter 5 arrival time:5 3 4 1 2
enter 5 burst time:5 4 4 3 7

p       A.T     B.T     C.T     TAT     WT
P4      1       3       4       3       0
P5      2       7       11      9       2
P2      3       4       15      12      8
P3      4       4       19      15      11
P1      5       5       24      19      14
average turnaround time is 11.600000
average wating timme is 7.000000
```

Figure1.1: Screenshot of output

Average Waiting Time: The average amount of time each process spends waiting in the ready queue before it gets CPU time. The average waiting time is 11.6000ms

Average Turnaround Time: The average amount of time taken for each process to complete execution, including both waiting time and CPU execution time. The average turnaround time is 7.0000ms.

The output typically includes a tabular representation of process details along with their waiting time and turnaround time. Additionally, the average waiting time and average turnaround time are displayed.

6. Conclusion

The FCFS scheduling algorithm is a simple yet fundamental CPU scheduling algorithm in operating systems. Although it has certain drawbacks, it provides insights into the basic principles of process scheduling. Through this lab experiment, we gained practical experience in implementing and evaluating the FCFS algorithm.