

1)

**Aim:** Write a Program to demonstrate First Come First Serve CPU scheduling**Program:**

```

#include<stdio.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]);
    }
    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;
            }
        }
    }
    ct[0]=at[0]+bt[0];
    for(i=1;i<n;i++)
    {

```

```

temp=0;
if(ct[i-1]<at[i])
{
    temp=at[i]-ct[i-1];
}
ct[i]=ct[i-1]+bt[i]+temp;
}
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\n",awt);
return 0;
}

```

**Output:**

```

sohail@LAPTOP-P1GHM93V:/mnt/e/college/5th sem/os lab$ gcc week5_1.c -o w51
sohail@LAPTOP-P1GHM93V:/mnt/e/college/5th sem/os lab$ ./w51
enter no of proccess you want:5
enter 5 process:5 4 3 2 1
enter 5 arrival time:3 2 5 1 4
enter 5 burst time:1 2 2 3 1

p      A.T    B.T    C.T    TAT    WT
P2     1       3      4      3      0
P4     2       2      6      4      2
P5     3       1      7      4      3
P1     4       1      8      4      3
P3     5       2     10     5      3
average turnaround time is 4.000000
average wating timme is 2.200000
sohail@LAPTOP-P1GHM93V:/mnt/e/college/5th sem/os lab$ █

```

2)

**Aim:** Write a Program to demonstrate Shortest Job First or Shortest Job next CPU scheduling

**Program:(preemptive)**

```
#include <stdio.h>
int main()
{
    int arrival_time[10], burst_time[10], temp[10];
    int i, smallest, count = 0, time, limit;
    double wait_time = 0, turnaround_time = 0, end;
    float average_waiting_time, average_turnaround_time;
    printf("\nEnter the Total Number of Processes:\t");
    scanf("%d", &limit);
    printf("\nEnter Details of %d Processes\n", limit);
    for(i = 0; i < limit; i++)
    {
        printf("\nEnter Arrival Time:\t");
        scanf("%d", &arrival_time[i]);
        printf("Enter Burst Time:\t");
        scanf("%d", &burst_time[i]);
        temp[i] = burst_time[i];
    }
    burst_time[9] = 9999;
    for(time = 0; count != limit; time++)
    {
        smallest = 9;
        for(i = 0; i < limit; i++)
        {
            if(arrival_time[i] <= time && burst_time[i] < burst_time[smallest] && burst_time[i] > 0)
            {
                smallest = i;
            }
        }
        burst_time[smallest]--;
        if(burst_time[smallest] == 0)
        {
            count++;
            end = time + 1;
            wait_time = wait_time + end - arrival_time[smallest] - temp[smallest];
            turnaround_time = turnaround_time + end - arrival_time[smallest];
        }
    }
    average_waiting_time = wait_time / limit;
    average_turnaround_time = turnaround_time / limit;
    printf("\n\nAverage Waiting Time:\t%lf\n", average_waiting_time);
    printf("Average Turnaround Time:%lf\n", average_turnaround_time);
    return 0;
}
```

**Output:**

```

PS E:\college\5th sem\os lab> cd "e:\college\5th sem\os lab\"
; if ($?) { gcc week5_2_1.c -o week5_2_1 } ; if ($?) { .\week5_2_1 }

```

Enter the Total Number of Processes: 3

Enter Details of 3 Processes

Enter Arrival Time: 1  
Enter Burst Time: 2

Enter Arrival Time: 2  
Enter Burst Time: 5

Enter Arrival Time: 3  
Enter Burst Time: 3

Average Waiting Time: 1.333333  
Average Turnaround Time:4.666667  
PS E:\college\5th sem\os lab> █

#### Program(non preemptive):

```

#include<stdio.h>
int main()
{
    int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
    float avg_wt,avg_tat;
    printf("Enter number of process:");
    scanf("%d",&n);

    printf("\nEnter Burst Time:n");
    for(i=0;i<n;i++)
    {
        printf("p%d:",i+1);
        scanf("%d",&bt[i]);
        p[i]=i+1;
    }

    //sorting of burst times
    for(i=0;i<n;i++)
    {
        pos=i;
        for(j=i+1;j<n;j++)
        {

```

```

        if(bt[j]<bt[pos])
            pos=j;
    }

    temp=bt[i];
    bt[i]=bt[pos];
    bt[pos]=temp;

    temp=p[i];
    p[i]=p[pos];
    p[pos]=temp;
}

wt[0]=0;

for(i=1;i<n;i++)
{
    wt[i]=0;
    for(j=0;j<i;j++)
        wt[i]+=bt[j];

    total+=wt[i];
}

avg_wt=(float)total/n;
total=0;

printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");
for(i=0;i<n;i++)
{
    tat[i]=bt[i]+wt[i];
    total+=tat[i];
    printf("\np%d\t\t %d\t\t %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
}

avg_tat=(float)total/n;
printf("\n\nAverage Waiting Time=%f",avg_wt);
printf("\n\nAverage Turnaround Time=%f\n",avg_tat);
}

```

### Output:

```

PS E:\college\5th sem\os lab> cd "e:\college\5th sem\os lab\"; if ($?) { gcc week5_2.2.c -o week5_2.2 }; if ($?) { .\week5_2.2 }
Enter number of process:5
Enter Burst Time:
p1:3
p2:2
p3:4
p4:3
p5:7
Process    Burst Time    Waiting Time    Turnaround Time
p2         2             0              2
p1         3             2              5
p4         3             5              8
p3         4             8             12
p5         7            12             19
Average Waiting Time=5.400000
Average Turnaround Time=9.200000
PS E:\college\5th sem\os lab>

```

```
wait time = wait time + total - arrival time[i] - burst time[i];
```

```

        turnaround_time = turnaround_time + total - arrival_time[i];
        counter = 0;
    }
    if(i == limit - 1)
    {
        i = 0;
    }
    else if(arrival_time[i + 1] <= total)
    {
        i++;
    }
    else
    {
        i = 0;
    }
}

average_wait_time = wait_time * 1.0 / limit;
average_turnaround_time = turnaround_time * 1.0 / limit;
printf("\n\nAverage Waiting Time:\t%f", average_wait_time);
printf("\nAvg Turnaround Time:\t%f\n", average_turnaround_time);
return 0;
}

```

#### Output:

```

PS E:\college\5th sem\os lab> cd "e:\college\5th sem\os lab\"
; if ($?) { gcc week5_3.c -o week5_3 } ; if ($?) { .\week5_3
} ; if ($?) { gcc week5_3.c -o week5_3 } ; if ($?) { .\week5_3 }

Enter Total Number of Processes: 3

Enter Details of Process[1]
Arrival Time: 1
Burst Time: 4

Enter Details of Process[2]
Arrival Time: 3
Burst Time: 6

Enter Details of Process[3]
Arrival Time: 1
Burst Time: 5

Enter Time Quantum: 2

Process ID      Burst Time      Turnaround Time      Waiting Time
Process[1]      4              3                   -1
Process[2]      6              11                  5
Process[3]      5              14                  9

Average Waiting Time: 4.333333
Avg Turnaround Time: 9.333333
PS E:\college\5th sem\os lab>

```

#### 4)

**Aim:** Write a Program to demonstrate Priority CPU Scheduling

**Program(preemptive):**

```
#include<stdio.h>
```

```
struct process
```