

Experiment No : 09

Experiment Name: Implementation of Priority Scheduling Algorithm.

Objectives:

- What is Priority Scheduling Algorithm?
- How to implement in c?

❖ Priority Scheduling Algorithm :

What is priority scheduling algorithm.

→ Processes with same priority are execute on first come first served (FCFS) basis.

→ Priority scheduling is a non-preemptive algorithm and one of most common scheduling algorithm in batch systems.

→ Each process is assigned a priority process with highest priority is to be execute first and so on.

Aim : To write a c program to implement the CPU scheduling priority algorithm.

Description:

To calculate the average waiting time in the priority algorithm, sort the burst times according to their priorities and then calculate the average waiting time of the processes. The waiting time of each process is obtained by summing up the burst times of all the previous processes.

Algorithm:

Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Sort the ready queue according to the priority number.

Step 5: Set the waiting of the first process as 0 and its burst time as its turnaround time

Step 6: Arrange the processes based on process priority

Step 7: for each process in the Ready Q calculate

a) Waiting time(n)= waiting time (n-1) + Burst time (n-1)

b) Turnaround time (n)= waiting time(n)+Burst time(n)

Step 8: Calculate

c) Average waiting time = Total waiting Time / Number of process

d) Average Turnaround time = Total Turnaround Time / Number of process
Print the results in an order

Step 9: Stop the process

Source Code:

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

```
void main()
```

```
{
```

```
    int n,i,j,pr[30],p[30],bt[30],temp,c,wt[30],tat[30];
```

```
    float awt,atat;
```

```
    printf("Enter the num of process : ");
```

```
    scanf("%d",&n);
```

```
    for(i=0; i<n; i++)
```

```
    {
```

```
        pr[i]=p[i]=i;
```

```
        printf("\nEnter the burst time & priority for process P%d = ",i);
```

```
        scanf("%d%d",&bt[i],&p[i]);
```

```
    }
```

```
    for(i=0; i<n; i++)
```

```
    {
```

```
        for(j=i+1; j<n; j++)
```

```
        {
```

```
            if(p[i]>p[j])
```

```
            {
```

```
                temp=p[i];
```

```
                p[i]=p[j];
```

```
                p[j]=temp;
```

```
                temp=pr[i];
```

```
                pr[i]=pr[j];
```

```
                pr[j]=temp;
```

```

        temp=bt[i];
        bt[i]=bt[j];
        bt[j]=temp;

    }
}

printf("\n\tprocess \t priority \t burst time \t waiting time \t turnaround
time\n");

wt[0]=awt=0;
tat[0]=atat=bt[0];
for(i=1; i<n; i++)
{
    wt[i]=wt[i-1]+bt[i-1];
    tat[i]=tat[i-1]+bt[i];

    awt=awt+wt[i];
    atat= atat+tat[i];
}
for(i=0; i<n; i++)
{
    printf("\tP%d \t\t %d \t\t %d \t\t %d \t\t %d \t\t
%d\n",pr[i],p[i],bt[i],wt[i],tat[i]);
}
awt=awt/n;
atat=atat/n;
printf("\nAverage waiting time %.2f\n",awt);
printf("Average turnaround time %.2f\n",atat);
return 0;
}

```

Output:

```
E:\programming\OS_Lab\Priority_Algorithm.exe
Enter the num of process : 5
Enter the burst time & priority for process P0 = 10 3
Enter the burst time & priority for process P1 = 1 1
Enter the burst time & priority for process P2 = 2 4
Enter the burst time & priority for process P3 = 1 5
Enter the burst time & priority for process P4 = 5 2

process      priority    burst time    waiting time    turnaround time
P1           1           1             0              1
P4           2           5             1              6
P0           3          10             6             16
P2           4           2            16             18
P3           5           1            18             19

Average waiting time 8.20
Average turnaround time 12.00

Process returned 30 (0x1E)   execution time : 43.617 s
Press any key to continue.
```

Result:

Priority scheduling program was executed and verified successfully.