

Name : Masuk Mia

ID : IT-18049

Lab report no. : 09

Lab report name : Lab report on implementation of Priority scheduling algorithm.

Aim and objectives: To learn about Priority scheduling algorithm , implement it with a c program, to learn how to use this algorithm.

Explanation:

i) **Priority scheduling algorithm:** Priority Scheduling is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority. Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems. This algorithm provides a good mechanism where the relative important of each process may be precisely defined.

ii) **Implementation of Priority scheduling algorithm in C Program:**

```
Code:    #include<stdio.h>    int    main()    {                                int
bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg_wt,avg_tat;

printf("Enter Total Number of Process:");    scanf("%d",&n);

    printf("\nEnter Burst Time and Priority\n\n");

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

        {
```

```

        printf("Burst Time of P[%d] process:",i+1);
scanf("%d",&bt[i]);    printf("Priority of
P[%d] process:",i+1);    scanf("%d",&pr[i]);
p[i]=i+1;    printf("\n");

    }

```

```

    for(i=0;i<n;i++)
    {
        pos=i;
    for(j=i+1;j<n;j++)

        {
            if(pr[j]<pr[pos])
            pos=j;

        }
    }

```

```

        temp=pr[i];
pr[i]=pr[pos];
pr[pos]=temp;

```

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

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

```
    }
```

```
    wt[0]=0;
```

```
    //calculate waiting time
```

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

```
    {        wt[i]=0;
```

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

```
    wt[i]+=bt[j];
```

```
    total+=wt[i];
```

```
    }
```

```
    avg_wt=total/n;    //average waiting time
```

```
    total=0;
```

```

    printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");

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

    {      tat[i]=bt[i]+wt[i];    //calculate turnaround time
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=total/n;    //average turnaround time
printf("\n\nAverage Waiting Time=%d",avg_wt);
printf("\nAverage Turnaround Time=%d\n",avg_tat);

    return 0;

}

```

Output:

```
"E:\3-1\Operating system Lab\Zafrul_Hasan_Khan\Priority_scheduling_algorithm.exe"
Enter Total Number of Process:4
Enter Burst Time and Priority
Burst Time of P[1] process:23
Priority of P[1] process:11
Burst Time of P[2] process:8
Priority of P[2] process:21
Burst Time of P[3] process:13
Priority of P[3] process:19
Burst Time of P[4] process:22
Priority of P[4] process:25

Process      Burst Time      Waiting Time      Turnaround Time
P[1]         23              0                23
P[3]         13              23              36
P[2]         8               36              44
P[4]         22              44              66

Average Waiting Time=25
Average Turnaround Time=42

Process returned 0 (0x0)   execution time : 38.605 s
Press any key to continue.
```

Conclusion: From this lab , I have shown this algorithm easy to use as scheduling method. This algorithm executes the processes according to the priority process with higher priority is executed first. At the end , this algorithm will be helpful for using as scheduling method .