

Ex. No.: 6c)

Date: 27-02-25

### PRIORITY SCHEDULING

Aim:

To implement priority scheduling technique

Algorithm:

1. Get the number of processes from the user.
2. Read the process name, burst time and priority of process.
3. Sort based on burst time of all processes in ascending order based priority 4.
- Calculate the total waiting time and total turnaround time for each process 5.
- Display the process name & burst time for each process.
6. Display the total waiting time, average waiting time, turnaround time

Program Code:

```
#include <stdio.h>
int main () {
    int n;
    printf ("Enter the number of process : \n");
    scanf ("%d", &n);
    int p[n];
    int pr[n];
    int pt [n]; int wt [n]; int tat[n];
    printf ("Enter the burst time of process: \n");
    for (int i=0; i<n; i++)
    {
        p[i] = i+1;
        printf ("p[%d] \n", i+1);
        printf ("Burst time: ");
        scanf ("%d", &bt [i]);
    }
}
```

```
printf("Priority: ");  
scanf("%d", &pr[i]);
```

```
int total_wt = 0, total_tat = 0;
```

```
for(int i = 0; i < n - 1; i++) {
```

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

```
    {
```

```
        if (pr[i] > pr[j]) {
```

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

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

```
            bt[j] = temp;
```

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

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

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

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

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

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

```
        }
```

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

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

```
        wt[i] = wt[i-1] + bt[i-1];
```

```
    }
```

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

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

```
    }
```

```

for (int i = 0; i < n; i++) {
    total_wt = wt[i];
    total_tat += tat[i];
}

printf (" \n Process \t Burst Time \t Priority \t waiting time \t Turnaround Time \n");

for (int i = 0; i < n; i++) {
    printf ("P[%d] \t %d \t %d \t %d \t %d \n", p[i], bt[i], pr[i], wt[i], tat[i]);
}

printf (" \n Average waiting time: %.2f", total_wt / n);

printf (" \n Average turnaround time: %.2f", total_tat / n);
}

```

Process	Burst time	Priority	Waiting time (ms)	Turnaround time
P[2]	8	1	0	8
P[1]	6	2	8	14
P[3]	7	3	14	21



Sample Output:

```
C:\Users\admin\Desktop\Untitled1.exe
Enter Total Number of Process:4
Enter Burst Time and Priority
P[1]
Burst Time:6
Priority:3
P[2]
Burst Time:2
Priority:2
P[3]
Burst Time:14
Priority:1
P[4]
Burst Time:6
Priority:4
Process    Burst Time    Waiting Time    Turnaround Time
P[3]       14            0              14
P[2]       2            14             16
P[1]       6            16             22
P[4]       6            22             28
Average Waiting Time=13
Average Turnaround Time=20
```

Enter the no of processes : 3

P[1]:

Burst time : 6

Priority : 2

P[2]:

Burst time : 8

Priority : 1

P[3]:

Burst time : 7

Priority : 3

Average waiting time : 7.33

Average turnaround time : 14.33

Result:

Therefore CPU scheduling has been  
successfully scheduled using Priority  
Algorithm.