

Ex. No.: 6c)

Date:

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, ar; int pid[n], b[n], pr[n], w[n], tat[n];
    printf("Enter the no. of processes:");
    scanf("%d", &n);
    printf("Enter the arrival time:");
    scanf("%d", &ar);
    printf for (int i=0; i<n; i++) {
        printf("Enter burst time & priority %d:\n", i+1);

        pid[i] = i+1;
        printf("Burst time:");
        scanf("%d", &b[i]);
        printf("Priority:");
        scanf("%d", &pr[i]);
    }
}
```



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

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

```
        if (pr[j] > pr[j+1]) {
```

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

```
            pr[j] = pr[j+1];
```

```
            pr[j+1] = temp;
```

```
            int temp1 = b[j];
```

```
            b[j] = b[j+1];
```

```
            b[j+1] = temp1;
```

```
            int temp2 = pid[j];
```

```
            pid[j] = pid[j+1];
```

```
            pid[j+1] = temp2; }
```

```
        }
```

```
    }  
w[0] = 0;
```

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

```
        if (i == 0) {
```

```
            c[i] = b[i]; }
```

```
        else {
```

```
            c[i] = c[i-1] + b[i]; } }
```

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

```
        tar[i] = c[i] - ar; }
```



```

float sum1 = 0;
for (int i=0; i<n; i++){
    sum1 = sum1 + tat[i];
}

float avg_tat = sum1/n;
for (int i=0; i<n; i++){
    w[i] = tat[i] - b[i];
}

```

```

float sum2 = 0;
for (int i=0; i<n; i++){
    sum2 = sum2 + tat w[i];
}

```

```

float avg_w = sum2/n;

```

```

printf("In Process \t Burst \t priority \t Completion  

\t Waiting \t Turn Around \n");

```

```

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

```

```

    printf("%d \t %d \t %d \t %d \t %d \n",
        pid[i], b[i], pr[i], wait[i], tat[i]);

```

```

}

```

```

printf("In The Average waiting time is : %.1f \n",
    avg_w);

```

```

printf("In The Average turn Around time is : %.1f \n",
    avg_tat);

```

```

}

```


Input:

Enter the num of processes: 5

Enter Burst time & priority for process 1:

Burst time: 6

priority: 2

Enter for process 2:

Burst time: 12 priority: 4

Enter for process 3:

Burst time: 1 priority: 5

Enter for process 4:

Burst time: 3 priority: 1

Enter for process 5:

Burst time: 4 priority: 3

process	Burst	priority	Waiting	Turnaround
4	3	1	0	3
1	6	2	3	9
5	4	3	9	13
2	12	4	13	25
3	1	5	25	26

Average waiting time: 10.0 ms

Average Turnaround time: 15.2 ms.

Sample Output:

```
C:\Users\admin\Desktop\Untitled1.txt
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
```

Result:

Thus the priority scheduling algorithm
~~is~~ executed.

Signature