

Ex. No.: 6c)

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, i, j, p[20], bt[20], pr[20], wt[20], tat[20], temp;
    float avg_wt = 0, avg_tat = 0;

    printf("Enter total number of processes: ");
    scanf("%d", &n);

    for(i = 0; i < n; i++) {
        printf("Enter burst time and priority for process %d: ", i + 1);
        scanf("%d%d", &bt[i], &pr[i]);
        p[i] = i + 1;
    }

    for(i = 0; i < n - 1; i++) {
        for(j = i + 1; j < n; j++) {
            if(pr[i] > pr[j]) {
                temp = pr[i];
                pr[i] = pr[j];
                pr[j] = temp;

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

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

    for(i = 0; i < n; i++) {
        wt[i] = 0;
        tat[i] = 0;
        for(j = 0; j < i; j++) {
            wt[i] += bt[j];
            tat[i] += p[j];
        }
        avg_wt += wt[i];
        avg_tat += tat[i];
    }

    printf("Total waiting time: %f\n", avg_wt);
    printf("Average waiting time: %f\n", avg_wt/n);
    printf("Total turnaround time: %f\n", avg_tat);
    printf("Average turnaround time: %f\n", avg_tat/n);
}
```

```

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

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

for(i = 0; i < n; i++) {
    tat[i] = bt[i] + wt[i];
    avg_wt += wt[i];
    avg_tat += tat[i];
}

avg_wt /= n;
avg_tat /= n;

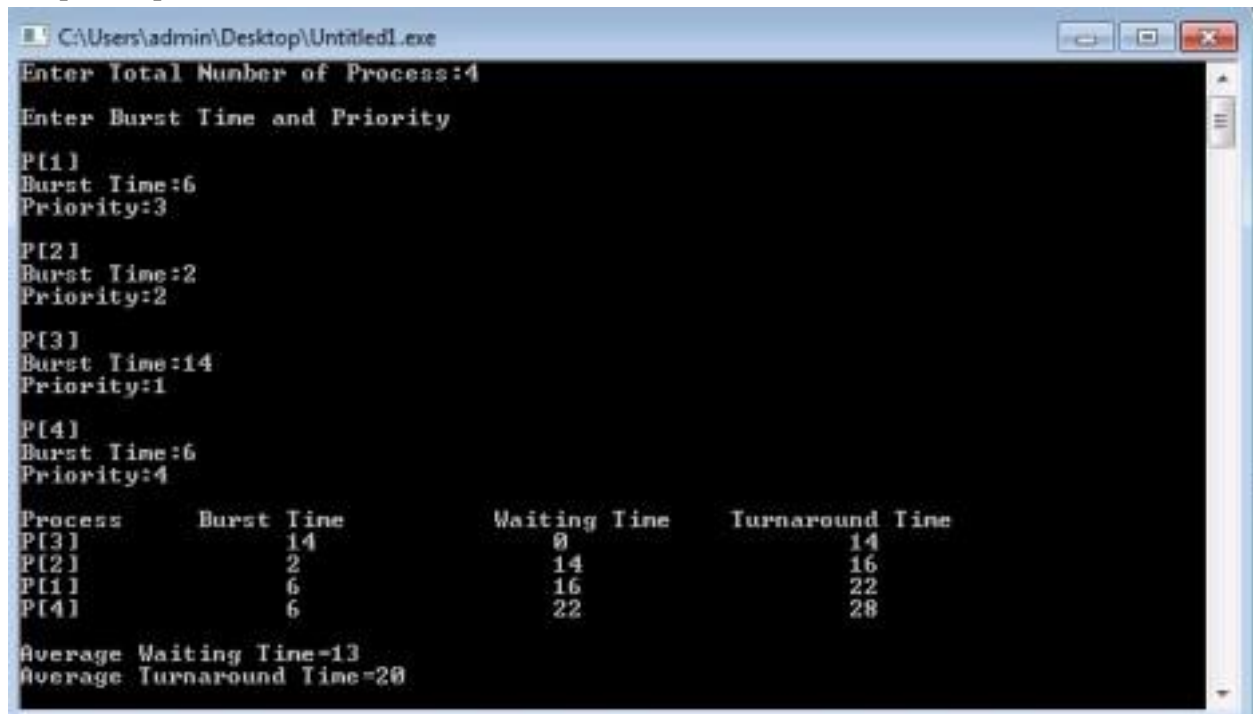
printf("\nProcess\tBurst Time\tPriority\tWaiting Time\tTurnaround Time\n");
for(i = 0; i < n; i++) {
    printf("P%d\t%d\t%d\t\t%d\t\t%d\n", p[i], bt[i], pr[i], wt[i], tat[i]);
}

printf("\nAverage Waiting Time = %.2f", avg_wt);
printf("\nAverage Turnaround Time = %.2f\n", avg_tat);

return 0;
}

```

Sample Output:



```
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      Priority      Waiting Time      Turnaround Time
P[3]          14              1              0              14
P[2]           2              2             14             16
P[1]           6              3             16             22
P[4]           6              4             22             28
Average Waiting Time=13
Average Turnaround Time=28
```

Output:

Enter total number of processes: 4
Enter burst time and priority for process 1: 10 3
Enter burst time and priority for process 2: 1 1
Enter burst time and priority for process 3: 2 4
Enter burst time and priority for process 4: 1 2

Process	Burst Time	Priority	Waiting Time	Turnaround Time
P2	1	1	0	1
P4	1	2	1	2
P1	10	3	2	12
P3	2	4	12	14

Average Waiting Time = 3.75
Average Turnaround Time = 7.25

Result:

The priority scheduling technique has been implemented successfully and the output has been verified.