

Date: 27/02/25

Aim: To implement priority scheduling technique

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.
4. Calculate the total waiting time and total turnaround time for each process 5.
5. Display the process name & burst time for each process.
6. Display the total waiting time, average waiting time, turnaround time

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

```
int main() {
```

```
int n;
```

```
printf("Enter no. of process\n");
```

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

int P[n], bt[n], wt[n], tat[n], pr[n], t1, t2, t3.

float sum1 = 0, sum2 = 0;

```
printf ("enter process no, BT & priority\n");
```

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

Scan for "1. d 1. d 1. d", & P[i], & bt[i], & ps[i]).

2

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

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

if $\underline{CP_n [j]} > M_n [j+1] \} \{$

$$t_1 = p_2 [j];$$
$$p_n[j] = p_n[j+1];$$
$$p_{j+1} [j+1] = t_1;$$


```

    t2 = p[j];
    p[j] = p[j+1];
    p[j+1] = t2;
    t3 = b[j];
    b[j] = b[j+1];
    b[j+1] = t3;
}
}
}
wt[0] = 0;
for (int i=1; i < n; i++) {
    wt[i] = wt[i-1] + c[i-1];
    s1 += wt[i];
}
for (int i=0; i < n; i++) {
    tat[i] = wt[i] + b[i];
    s2 += tat[i];
}
for (int i=0; i < n; i++) {
    printf("P %.1f B %.1f T %.1f W %.1f",
           p[i], b[i], pr[i], wt[i], tat[i]);
}
printf("\n avg wt = %.2f", s1/n);
printf("\n avg tat = %.2f", s2/n);
}

```


OUTPUT:

enter n

4

enter process no, BT priority

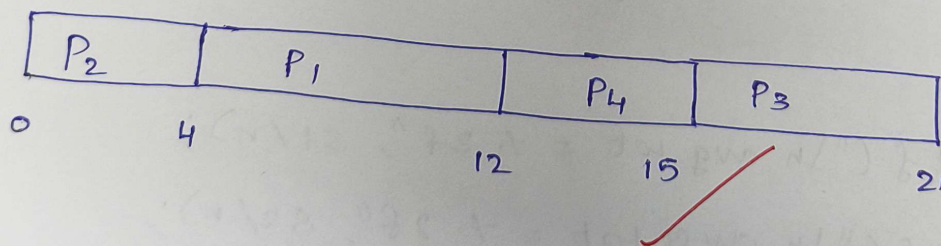
1	8	2
2	4	1
3	6	4
4	3	3

Process	BT	Priority	WT	TAT
P ₂	4	1	0	4
P ₁	8	2	4	12
P ₄	3	3	12	15
P ₃	6	4	15	21

avg. WT = 7.75 ms

avg. tat = 13.00 ms

Gantt Chart:




```
Sample Output:
CAUsers\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
```

Result:

Hence the priority scheduling is implemented
and Executed successfully.

[Handwritten signature]