

Ex. No.: 6c)

Date: 27/2/25

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>
#include <stdlib.h>
int main()
{
    int n;
    printf("Enter no of processes: \n");
    scanf("%d", &n);
    int bt[n], p[n], ct[n], at = 0, tat[n], wt[n], aat = 0, awt = 0;
    printf("Enter the burst time: \n");
    for (int i = 0; i < n; i++)
        scanf("%d", &bt[i]);
    printf("Enter priority of processes: \n");
    for (int i = 0; i < n; i++)
        scanf("%d", &p[i]);
    int sp[n];
    for (int i = 0; i < n; i++)
        sp[i] = p[i];
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - 1 - i; j++)
        {
            if (sp[j + 1] < sp[j])
            {
                swap(&sp[j], &sp[j + 1]);
            }
        }
    }
}
```



```

int temp = sp[i+1];
sp[i+1] = sp[j];
sp[j] = temp;
}
}

```

```

int c = 0;
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        if (sp[i] == p[j])
        {
            ct[j] = ct[j] + b[ct[j]];
            c = ct[j];
            tat[j] = ct[j] - at;
            wt[j] = tat[j] - bt[j];
        }
    }
}

```

```

printf("In Completion Time\n");
for (int i = 0; i < n; i++)
    printf("%d\n", ct[i]);
printf("\n Turn around time\n");
for (int i = 0; i < n; i++)
    printf("%d\n", tat[i]);
printf("\n Wait time\n");
for (int i = 0; i < n; i++)
    printf("%d\n", wt[i]);

```

```

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

```

```

printf("\n Average Turn around time : %.2f\n",
        (float) atat / n,
        (float) awt / n);

```

Output:

Enter no of processes: 4

Enter the burst time:

13

5

8

4

Enter priority of process:

3

2

4

1

Completion time:

22

9

30

4

Turn around time:

22

9

30

4

wait time:

9

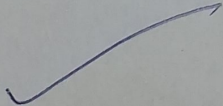
4

22

0

Average Turn around time: 16.25ms

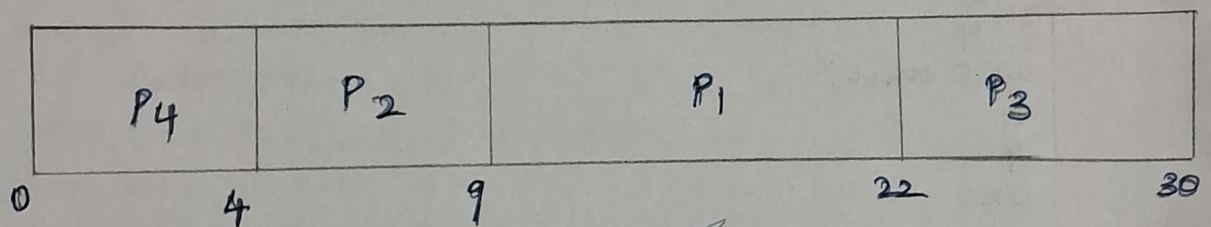
Average wait time: 8.75ms.



Process	BST (ms)	Priority	CT (ms)	TAT (ms)	WT (ms)
P ₁	13	3	22	22	9
P ₂	5	2	9	9	4
P ₃	8	4	30	30	22
P ₄	4	1	4	4	0

Average turn around time: 16.25 ms

Average wait time: 8.75 ms



Sample Output:

```
CAUser\admin\Desktop\Untitled1.c
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: The priority scheduling technique is implemented using C.

