Ex No.: 6c PRIORITY SCHEDULING

Date : 06.03.2025

Aim :

To implement Priority scheduling technique.

Code:

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number of processes: ");
    scanf("%d", &n);

    int at[n], bt[n], priority[n], wt[n], tat[n], ct[n], id[n];

    for (int i = 0; i < n; i++) {
        id[i] = i + 1;
        printf("Process %d Arrival Time: ", i + 1);
        scanf("%d", &at[i]);
        printf("Process %d Burst Time: ", i + 1);
        scanf("%d", &bt[i]);
        printf("Process %d Priority (lower value = higher priority): ", i + 1);
        scanf("%d", &priority[i]);
    }

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

                temp = at[j]; at[j] = at[j + 1]; at[j + 1] = temp;
                temp = bt[j]; bt[j] = bt[j + 1]; bt[j + 1] = temp;
                temp = priority[j]; priority[j] = priority[j + 1]; priority[j + 1] = temp;
                temp = id[j]; id[j] = id[j + 1]; id[j + 1] = temp;
            }
        }
    }

    int time = 0, completed = 0;
    while (completed < n) {
        int start = completed, end = completed;
        while (end < n && at[end] <= time)
            end++;
```

```
        for (int i = start; i < end - 1; i++) {
            for (int j = start; j < end - i - 1; j++) {
                if (priority[j] > priority[j + 1]) {
                    int temp;

                    temp = at[j]; at[j] = at[j + 1]; at[j + 1] = temp;
                    temp = bt[j]; bt[j] = bt[j + 1]; bt[j + 1] = temp;
                    temp = priority[j]; priority[j] = priority[j + 1]; priority[j + 1] = temp;
                    temp = id[j]; id[j] = id[j + 1]; id[j + 1] = temp;
                }
            }
        }

        time = (time < at[completed]) ? at[completed] : time;
        ct[completed] = time + bt[completed];
        tat[completed] = ct[completed] - at[completed];
        wt[completed] = tat[completed] - bt[completed];
        time = ct[completed];
        completed++;
    }

    printf("\nProcess\tAT\tBT\tPriority\tCT\tTAT\tWT\n");
    float totalTAT = 0, totalWT = 0;
    for (int i = 0; i < n; i++) {
        printf("%d\t%d\t%d\t%d\t\t%d\t%d\t%d\n", id[i], at[i], bt[i], priority[i], ct[i], tat[i],
wt[i]);
        totalTAT += tat[i];
        totalWT += wt[i];
    }

    printf("\nAverage Turnaround Time: %.2f", totalTAT / n);
    printf("\nAverage Waiting Time: %.2f\n", totalWT / n);

    return 0;
}
```

Output:

```
Enter the number of processes: 3
Process 1 Arrival Time: 0
Process 1 Burst Time: 7
Process 1 Priority (lower value = higher priority): 2
Process 2 Arrival Time: 0
Process 2 Burst Time: 3
Process 2 Priority (lower value = higher priority): 3
Process 3 Arrival Time: 0
Process 3 Burst Time: 10
Process 3 Priority (lower value = higher priority): 1

Process AT      BT      Priority        CT      TAT     WT
3       0       10      1               10      10      0
1       0       7       2               17      17      10
2       0       3       3               20      20      17

Average Turnaround Time: 15.67
Average Waiting Time: 9.00
```

Result:

Thus the implement Priority scheduling technique has been executed successfully.