Ex No.: 6b   SHORTEST JOB FIRST SCHEDULING

Date : 26.02.2025

Aim :

To implement Shortest Job First (SJF) scheduling technique.

Code:

```c
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    int pro[n], bt[n], at[n], wt[n], tat[n], ct[n];
    float total_wt = 0, total_tat = 0;
    printf("Enter the Burst time and Arrival time : \n");

    for(int i = 0; i < n; i++)
    {
        printf("Process %d Burst time: ", i + 1);
        scanf("%d", &bt[i]);
        printf("Process %d Arrival time: ", i + 1);
        scanf("%d", &at[i]);
        pro[i] = i + 1;
    }

    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (at[j] > at[j + 1] || (at[j] == at[j + 1] && bt[j] > bt[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 = pro[j];
                pro[j] = pro[j + 1];
                pro[j + 1] = temp;
            }
        }
    }
```

```c
    ct[0] = at[0] + bt[0];
    for(int i = 1; i < n; i++)
    {
        if (ct[i - 1] < at[i])
            ct[i] = at[i] + bt[i];
        else
            ct[i] = ct[i - 1] + bt[i];
    }

    for(int i = 0; i < n; i++)
    {
        tat[i] = ct[i] - at[i];
        total_tat += tat[i];
    }

    for(int i = 0; i < n; i++)
    {
        wt[i] = tat[i] - bt[i];
        total_wt += wt[i];
    }

    float avg_wt = total_wt / n;
    float avg_tat = total_tat / n;

    printf("Processes   Arrival time   Burst time   Completion time   Turn around time
Waiting time\n");
    for (int i = 0; i < n; i++)
    {
        printf("   %d        %d          %d          %d           %d            %d\n", pro[i], at[i],
bt[i], ct[i], tat[i], wt[i]);
    }

    printf("Average waiting time = %.2f\n", avg_wt);
    printf("Average turn around time = %.2f\n", avg_tat);

    return 0;
}
```

Output:

```
Enter the number of processes: 5
Enter the Burst time and Arrival time :
Process 1 Burst time: 10
Process 1 Arrival time: 0
Process 2 Burst time: 1
Process 2 Arrival time: 0
Process 3 Burst time: 2
Process 3 Arrival time: 0
Process 4 Burst time: 1
Process 4 Arrival time: 0
Process 5 Burst time: 5
Process 5 Arrival time: 0
Processes    Arrival time    Burst time    Completion time    Turn around time    Waiting time
   2             0              1                1                  1                  0
   4             0              1                2                  2                  1
   3             0              2                4                  4                  2
   5             0              5                9                  9                  4
   1             0             10               19                 19                  9
Average waiting time = 3.20
Average turn around time = 7.00
```

Result:

Thus the implement Shortest Job First (SJF) scheduling technique has been executed successfully.