

Ex. No.: 6b)

Date: 22/12/25

SHORTEST JOB FIRST

Aim:

To implement the Shortest Job First (SJF) scheduling technique


Algorithm:

1. Declare the structure and its elements.
2. Get number of processes as input from the user.
3. Read the process name, arrival time and burst time
4. Initialize waiting time, turnaround time & flag of read processes to zero.
5. Sort based on burst time of all processes in ascending order
6. Calculate the waiting time and turnaround time for each process.
7. Calculate the average waiting time and average turnaround time.
8. Display the results.

Program Code:

```
#include <stdio.h>

int main()
{
    int n, i, j;
    printf("Enter the no of Process:");
    scanf("%d", &n);
    int bt[n], wt[n], tat[n], p[n], temp;
    float total_wt = 0, total_tat = 0;
    printf("Enter the burst time of the process \n");
    for (i = 0; i < n; i++) {
        scanf("%d", &bt[i]);
    }
}
```



```

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

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

```

wt[0] = 0

```

for (i=1; i<n; i++) {

```

```

    wt[i] = wt[i-1] + bt[i-1];
}

```

```

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

```

```

    tat[i] = bt[i] + wt[i];
}

```

printf ("In Process \t Bursttime \t Waiting time \t Turnaround\n");

```

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

```

```

    printf ("%d \t %d \t %d \t %d \n", p[i], bt[i],
            wt[i], tat[i]);
}

```

total_wt += wt[i];

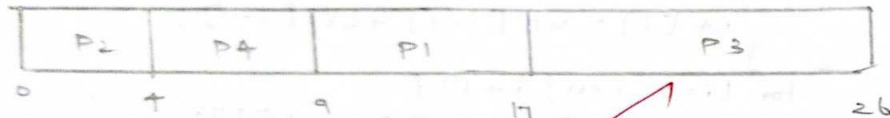
total_tat = tat[i];

```

}
printf ("In average wt : %.1f \n", total_wt/n);
printf ("average tat : %.1f \n", total_tat/n);
}

```

Gantt chart



Sample Output:

Enter the number of process:

4

Enter the burst time of the processes:

8 4 9 5

Process	Burst Time	Waiting Time	Turn Around Time
2	4	0	4
4	5	4	9
1	8	9	17
3	9	17	26

Average waiting time is: 7.5

Average Turn Around Time is: 13.0

Output:

Enter the number of Process : 4

Enter the burst time of process

8 4 9 5

Process	Burst time	Waiting time	Turnaround time
2	4	0	4
4	5	4	9
1	8	9	17
3	9	17	26

Result:

Hence the Shortest Job Scheduling is executed successfully.

Q. U. E.