

## Operating System – CS23431

<b>Ex 6b)</b>	<b>Shortest Job First</b>
<b>Name: B M Madhumitha</b>	
<b>Reg No: 230701168</b>	

**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>
#include<stdlib.h>
void sort(int bt[],int at[],int n,int process_id[]){
for(int i=0;i<n-1;i++){
for(int j=i+1;j<n;j++){
if(bt[i]>bt[j]){
int temp = bt[i];
bt[i] = bt[j];
bt[j] = temp;
temp = at[i];
at[i] = at[j];
at[j] = temp;
temp = process_id[i];
process_id[i] = process_id[j];
process_id[j] = temp;
}
}
}
}
void completiontime(int bt[],int at[],int n,int ct[]){
int min= at[0] , min_index = 0;
for(int i=1;i<n;i++){
if(min > at[i]){
min = at[i];
min_index = i;
}
}
}
```

```

int flag[n] ;
for(int i=0;i<n;i++){
    flag[i] =0;
}

flag[min_index]=1;
ct[min_index] = at[min_index]+bt[min_index];
int prev = min_index;
for(int j=0;j<n;j++){
    if(j==min_index) continue;
    int temp =0;
    for(int i = 0;i<n;i++){
        if((flag[i]==0)&&(at[i]<=ct[prev])){
            ct[j] = ct[prev]+bt[i];
            temp = 1;
            flag[i]=1;
            prev = j;
            break;
        }
    }
}

if(temp ==0){
    for(int i =0 ;i<n;i++){
        if(flag[i]==0){
            ct[j] = at[i] + bt[i];
            flag[i]=1;
            prev = j;
            break;
        }
    }
}
}
}

void turnAroundtime(int n, int ta[],int ct[],int at[]){
    for(int i=0;i<n;i++){
        ta[i] = ct[i] - at[i];
    }
}

void waitingtime(int n, int ta[],int bt[],int wt[]){
    for(int i=0;i<n;i++){
        wt[i] = ta[i] - bt[i];
    }
}

void averagecal(int n,int ta[],int wt[]){
    int total_tat =0;
    int total_wt =0;
    for(int i=0;i<n;i++){
        total_tat += ta[i];
        total_wt += wt[i];
    }
}

```

```

    printf("The Average TurnAround Time: %.2f\nThe Average Waiting Time:
%.2f", (float)total_tat/n, (float)total_wt/n);
}
int main(){
int n;
printf("Enter the number of processes :");
scanf("%d", &n);
int bt[n], at[n], process_id[n];
for(int i=0; i<n; i++){
    printf("Enter the Processes %d's Arrival Time and Burst Time:", i+1);
    scanf("%d", &at[i]);
    scanf("%d", &bt[i]);
    process_id[i] = i+1;
}
int wt[n], ta[n], ct[n];
sort(bt, at, n, process_id);
completiontime(bt, at, n, ct);
turnAroundtime(n, ta, ct, at);
waitingtime(n, ta, bt, wt);
printf("Process_ID  Arrival time  Burst Time  Completion Time  TurnAround Time  Waiting Time\n");
printf("-----\n");

for(int i=0; i<n; i++){
printf(" %d      %d      %d      %d      %d      %d  \n", process_id[i], at[i], bt[i], ct[i], ta
[i], wt[i]);
}
averagecal(n, ta, wt);
return 0;

}

```

### Sample Output:

```

C:\Users\kambm\OneDrive\Desktop\Madhumitha\sem IV\OS Assignment\Final version>gcc SJF_FINAL.c -o sjf.exe
C:\Users\kambm\OneDrive\Desktop\Madhumitha\sem IV\OS Assignment\Final version>sjf.exe
Enter the number of processes :4
Enter the Processes 1's Arrival Time and Burst Time:2 4
Enter the Processes 2's Arrival Time and Burst Time:4 5
Enter the Processes 3's Arrival Time and Burst Time:1 8
Enter the Processes 4's Arrival Time and Burst Time:3 9
Process_ID  Arrival time  Burst Time  Completion Time  TurnAround Time  Waiting Time
-----
1           2           4           13              11              7
2           4           5           18              14              9
3           1           8           9               8              0
4           3           9           27              24              15
The Average TurnAround Time: 14.25
The Average Waiting Time: 7.75
C:\Users\kambm\OneDrive\Desktop\Madhumitha\sem IV\OS Assignment\Final version>

```

Result : Thus, the program was executed successfully.