Lakhan Kumawat - 1906055

Shortest Job First

Shortest job first(SJF) is a scheduling algorithm,

that is used to schedule processes in an operating system.

It is a very important topic in Scheduling when compared to

round-robin and FCFS Scheduling.

There are two types of SJF

Preemptive SJF

Non-Preemptive SJF

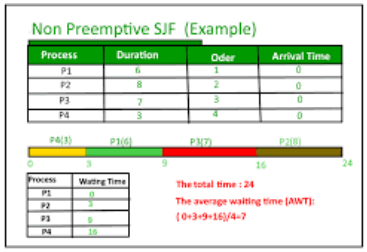
These algorithms schedule processes in the order in which the shortest job is done first. It has a minimum average waiting time.

There are 3 factors to consider while solving SJF, they are

Burst Time
Average waiting time
Average turnaround time

Non-Preemptive Shortest Job First

Here is an example

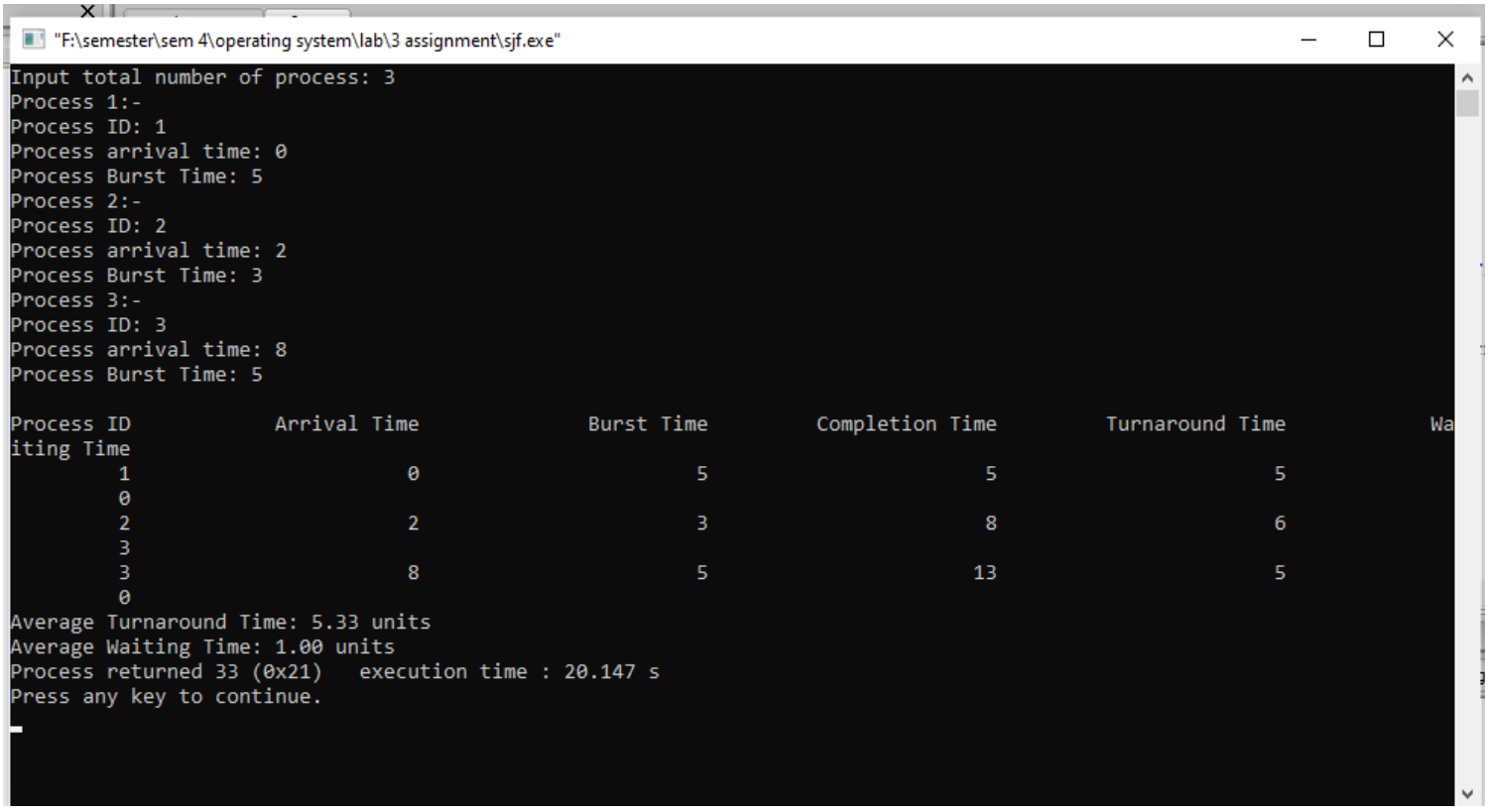


Lakhan Kumawat - 1906055

```
#include <stdio.h>
typedef struct Process Process;
struct Process{
char id[16];
int arrival,burst,completion,waiting,turnaround;
};
void Merge(Process A[],int l,int mid,int h){
int i=l,j=mid+1,k=l; Process B[100]; while(i<=mid && j<=h){
if(A[i].arrival<=A[j].arrival){ B[k++]=A[i++];
}
else{
}
}
B[k++]=A[j++];
while(i<=mid){
B[k++]=A[i++];
}
while(j<=h){
B[k++]=A[j++];
}
for(i=l;i<=h;i++){
A[i]=B[i];
}
}
void MergeSort(Process A[],int l,int h){
if(l<h){
int mid=(l+h)/2; MergeSort(A,l,mid); MergeSort(A,mid+1,h); Merge(A,l,mid,h);
}
}
void print(Process p[],int n){
printf("\n%10s\t%18s\t%18s\t%18s\t%18s\t%18s\t","Process ID","Arrival Time","Burst Time","Completion
Time","Turnaround Time","Waiting Time");
for(int i=0;i<n;i++){
printf("\n%10s\t%18d\t%18d\t%18d\t%18d\t%18d",p[i].id,p[i].arrival,p[i].burst,p[i].completion,p[i].turnaroun
d,p[i].waiting);
}
}
void SJF(Process p[],int n){
int arrived[n],min_burst_i=0;
int t=0;
float avg_t=0,avg_w=0; MergeSort(p,0,n-1); for(int i=0;i<n;i++){
arrived[i]=0;
}
for(int i=0;i<n;i++){
if(t<p[i].arrival){
t=p[i].arrival;
}
for(int  j=0;j<n;j++){
if(t>=p[j].arrival && arrived[j]!=-1){ //-1 for completed arrived[j]=1; //1 for arrival
min_burst_i=j;
}
if(p[min_burst_i].burst>p[j].burst && arrived[j]==1){ min_burst_i=j;
}
}
```

```
t=t+p[min_burst_i].burst; p[min_burst_i].completion=t;
p[min_burst_i].turnaround=p[min_burst_i].completion-p[min_burst_i].arrival;
p[min_burst_i].waiting=p[min_burst_i].turnaround-p[min_burst_i].burst; arrived[min_burst_i]=-1;
avg_t=avg_t+p[min_burst_i].turnaround; avg_w=avg_w+p[min_burst_i]. waiting;
}
print(p,n);
printf("\nAverage Turnaround Time: %.2f units",avg_t/n);
printf("\nAverage Waiting Time: %.2f units",avg_w/n);
}
void main(){
int n;
printf("Input total number of process: ");
scanf("%d",&n); Process p[n];
for(int i=0;i<n;i++){
printf("Process %d:-",i+1); printf("\nProcess ID: "); scanf("%s",&p[i].id); printf("Process arrival time: ");
scanf("%d",&p[i].arrival); printf("Process Burst Time: "); scanf("%d",&p[i].burst);
}
SJF(p,n);
}
```

OUTPUT



Explanations:-

Lakhan Kumawat - 1906055

In this program first I take the number of process then input process id decide which process come first then execute that program …

We know that in sjf

If a program is in ram but cpu execute another program and we know that sjf is primitive in nature then that program wait till another program which is execute (till competition)
Then we calculate within time

Turnaround time = burst time +waiting time of that program..

Average TAT is sum of total TAT by number of program
And
Average waiting is total waiting time by number of program