EXPERIMENT - 2

Operating Systems Lab

AIM

Write a program to implement CPU scheduling for the shortest job first.

Syeda Reeha Quasar
14114802719
6C7

EXPERIMENT – 2

Aim:

Write a program to implement CPU scheduling for shortest job first.

Theory:

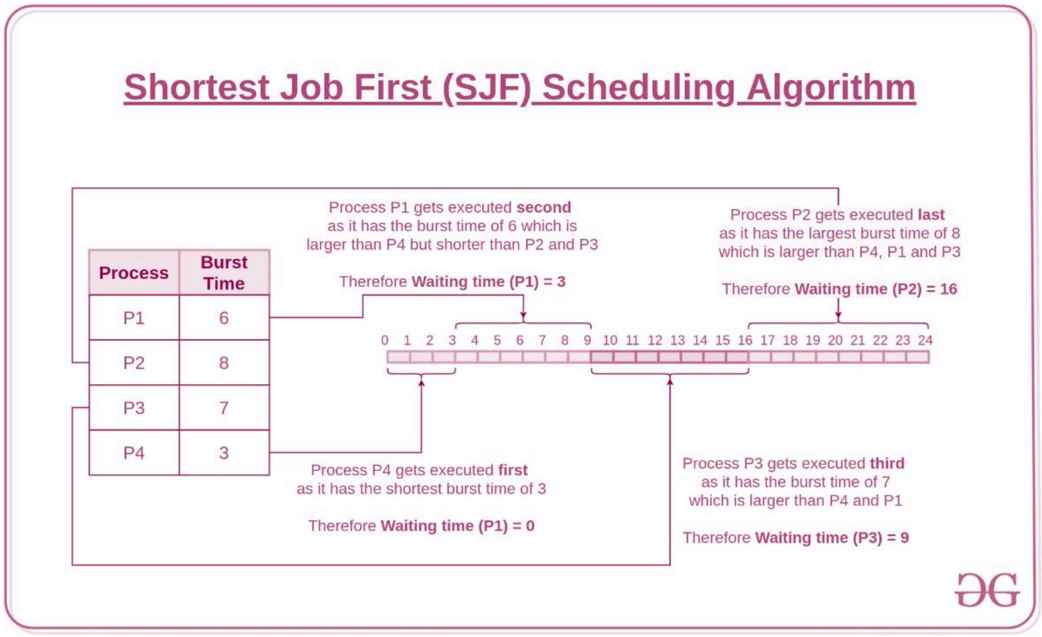
Shortest job first (SJF) or shortest job next, is a scheduling policy that selects the waiting process with the smallest execution time to execute next. SJN, also known as Shortest Job Next (SJN), can be preemptive or non-preemptive.

Characteristics of SJF Scheduling:

- Shortest Job first has the advantage of having a minimum average waiting time among all scheduling algorithms.
- It is a Greedy Algorithm.
- It may cause starvation if shorter processes keep coming. This problem can be solved using the concept of aging.
- It is practically infeasible as Operating System may not know burst times and therefore may not sort them. While it is not possible to predict execution time, several methods can be used to estimate the execution time for a job, such as a weighted average of previous execution times.
- SJF can be used in specialized environments where accurate estimates of running time are available.

Algorithm:

- Sort all the processes according to the arrival time.
- Then select that process that has minimum arrival time and minimum Burst time.
- After completion of the process make a pool of processes which after till the completion of the previous process and select that process among the pool which is having minimum Burst time.


Shortest Job First (SJF) Scheduling Algorithm
Process
Burst Time
P1 6
P2 8
P3 7
P4 3
Process P1 gets executed second as it has the burst time of 6 which is larger than P4 but shorter than P2 and P3
Therefore Waiting time (P1) = 3
Process P2 gets executed last as it has the largest burst time of 8 which is larger than P4, P1 and P3
Therefore Waiting time (P2) = 16
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Process P4 gets executed first as it has the shortest burst time of 3
Therefore Waiting time (P1) = 0
Process P3 gets executed third as it has the burst time of 7 which is larger than P4 and P1
Therefore Waiting time (P3) = 9

- Completion Time: Time at which process completes its execution.
- Turn Around Time: Time Difference between completion time and arrival time.
  Turn Around Time = Completion Time – Arrival Time

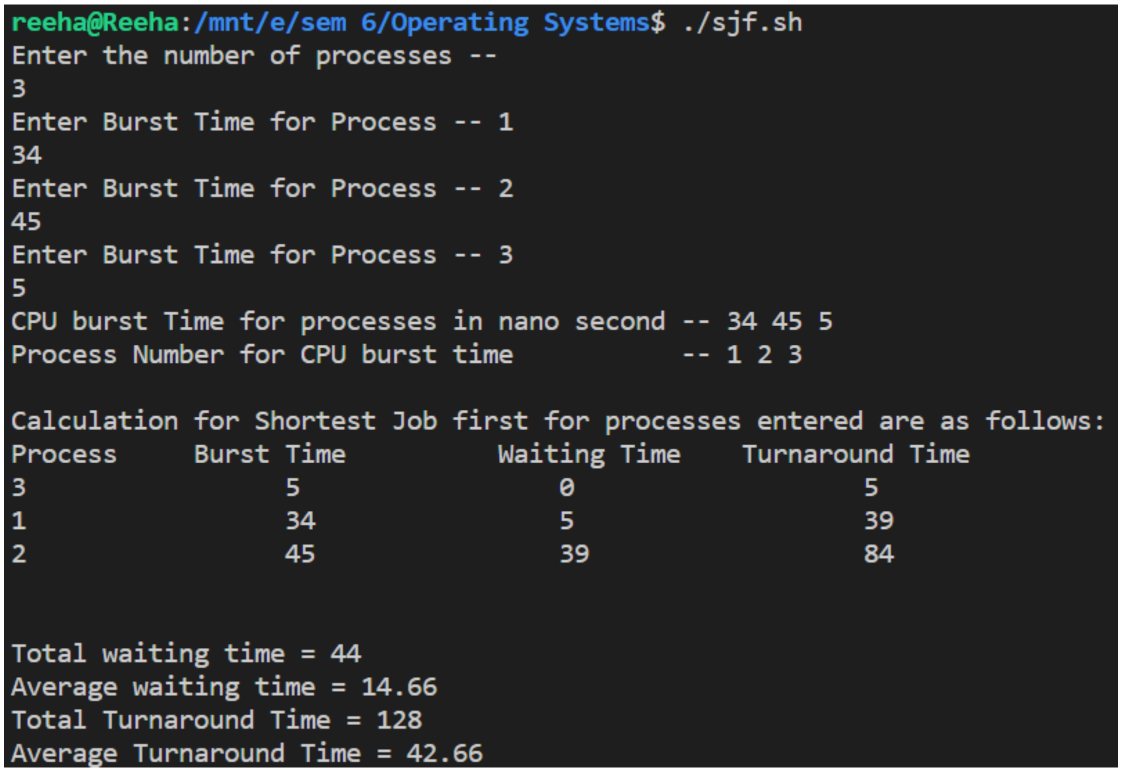
- Waiting Time(W.T): Time Difference between turn around time and burst time.
  Waiting Time = Turn Around Time – Burst Time

Source Code:

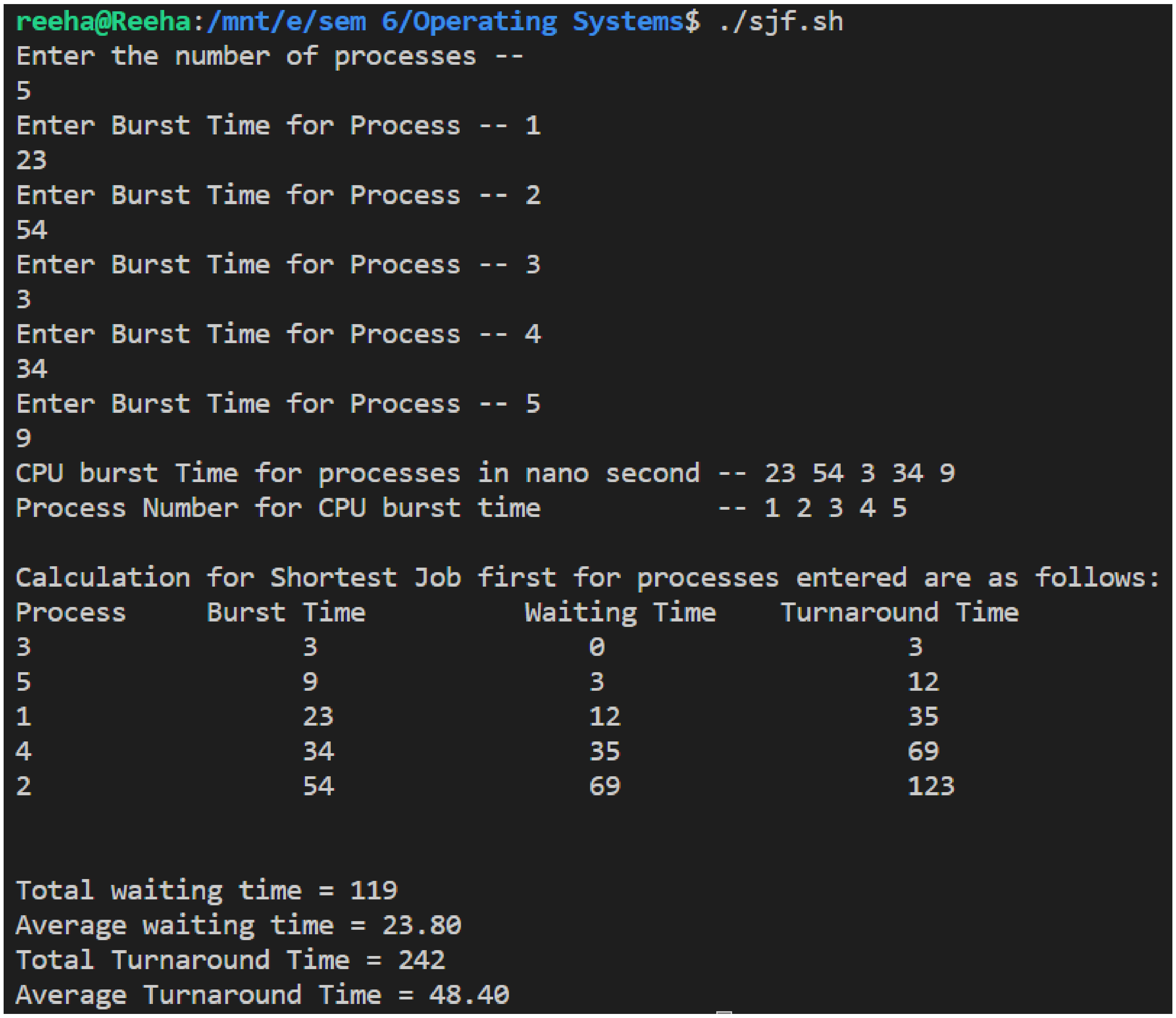
```
#! /bin/bash

function shortestjobfirst {
#Initializing Bash variables
awt=0
totalwt=0
totaltat=0
atat=0
temp=0
declare -a  wt
declare -a  tat
#sorting burst time in ascending order using selection sort
for ((i = 0; i<${number-1}; i++))
do

   for((j = 0; j<${number-i-1}; j++))
   do

      if [[ "${Btime[$j]}" -gt "${Btime[$j+1]}" ]];
      then
         # swaping Burst time array
         temp=${Btime[$j]};
         Btime[$j]=${Btime[$j+1]};
         Btime[$j+1]=$temp;

         #swaping process positon
         temp=${p[$j]};
         p[$j]=${p[$j+1]};
         p[$j+1]=$temp;
      fi
   done
done
   echo -e "Process\t    Burst Time    \tWaiting Time\tTurnaround Time"
   for ((i=1;i<=number;i++))
   do
      wt[i]=0;
      tat[i]=0;
      for ((j=0;j<i;j++))
      do
         wt[i]="$((wt[i]+Btime[j]))"     #calculate waiting time
      done

      totalwt="$((totalwt+wt[i]))"     #calculate total waiting time
      tat[i]="$((Btime[i]+wt[i]))"     #calculate turnaround time
      totaltat="$((totaltat+tat[i]))"     #calculate total turnaround time
      echo -e "${p[i]}\t\t  ${Btime[i]}\t\t    ${wt[i]}\t\t\t${tat[i]}"
   done
```

```
 awt=$(echo 'scale=2;' "$totalwt" / "$number" | bc -l)     #calculate average waiting time
 atat=$(echo 'scale=2;' "$totaltat" / "$number" | bc -l)   #calculate average turnaround time
echo -e "\n"
echo "Total waiting time =" "$totalwt"
echo "Average waiting time =" "$awt"
echo "Total Turnaround Time =" "$totaltat"
echo "Average Turnaround Time =" "$atat"
}

#Accepts user input for Number of Processes and Input Validation
echo "Enter the number of processes -- "
read -r number
while [[ "$number" -le 1 ]] || [[ -z "$number" ]]
do
echo "Error: Input valid number of processes or Input cannot be blank"
echo "Please try again."
echo "Enter the number of processes -- "
read -r number
done

declare -a Btime
declare -a  p
declare -a  rem_bt

#Accepts user input for Burst Time and Input Validation
for (( i=1; i<=number; i++ ))
do

echo "Enter Burst Time for Process -- $i"
read -r "Btime[i]"

while [[ "${Btime[i]}" -lt 1 ]] || [[ -z "${Btime[i]}" ]]
do
echo "Error: Input valid burst time for the process or Inputs cannot be blank"
echo "Please try again."
echo "Enter Burst Time for Process -- $i"
read -r "Btime[i]"
done
p[i]=$i  #contains process number
rem_bt[i]=${Btime[i]} #remaining process
done

echo -e "CPU burst Time for processes in nano second --" "${Btime[@]}"
echo -e "Process Number for CPU burst time          --" "${p[@]}"
echo ""
echo "Calculation for Shortest Job first for processes entered are as follows: "
shortestjobfirst
```

Output:



```
reeha@Reeha:/mnt/e/sem 6/Operating Systems$ ./sjf.sh
Enter the number of processes --
3
Enter Burst Time for Process -- 1
34
Enter Burst Time for Process -- 2
45
Enter Burst Time for Process -- 3
5
CPU burst Time for processes in nano second -- 34 45 5
Process Number for CPU burst time           -- 1 2 3

Calculation for Shortest Job first for processes entered are as follows:
Process      Burst Time          Waiting Time    Turnaround Time
3            5                   0                   5
1            34                  5                   39
2            45                  39                  84


Total waiting time = 44
Average waiting time = 14.66
Total Turnaround Time = 128
Average Turnaround Time = 42.66
```



```
reeha@Reeha:/mnt/e/sem 6/Operating Systems$ ./sjf.sh
Enter the number of processes --
5
Enter Burst Time for Process -- 1
23
Enter Burst Time for Process -- 2
54
Enter Burst Time for Process -- 3
3
Enter Burst Time for Process -- 4
34
Enter Burst Time for Process -- 5
9
CPU burst Time for processes in nano second -- 23 54 3 34 9
Process Number for CPU burst time           -- 1 2 3 4 5

Calculation for Shortest Job first for processes entered are as follows:
Process      Burst Time          Waiting Time    Turnaround Time
3            3                   0                   3
5            9                   3                   12
1            23                  12                  35
4            34                  35                  69
2            54                  69                  123


Total waiting time = 119
Average waiting time = 23.80
Total Turnaround Time = 242
Average Turnaround Time = 48.40
```