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:

```
n = int (input ("Enter the number of processes: "))

processes = []

bt = []
for i in range(n):
        P = int(input(f"Enter PID {i+1}: "))
        processes.append(p)


for i in range(n):
        b = int(input(f"Enter the BT for PID
                    {processes[i]}: "))

        bt.append(b)
# sorting process based on BT
for i in range(n):
        for j in range(i+1, n):
```

38

```python
if bt[i] > bt[j]:
        bt[i], bt[j] = bt[j], bt[i]
        processes[i], processes[j] = processes[j], processes[i]

ct = []
tat = []
wt = []
for i in range(n):
        if (i==0):
            ct.append(bt[i])
        else:
            ct.append(bt[i] + ct[i-1])

        tat.append(ct[i])       # calculate TAT
        wt.append(tat[i] - bt[i])  # calculate WT

sum_tat = sum(tat)
avg_tat = sum_tat /n
sum_wt = sum(wt)
avg_wt = sum_wt /n

print("PID\t BT \t CT \t TAT \t WT")
for i in range(n):
        print(f"{processes[i]}\t {bt[i]}\t {ct[i]}\t
        {tat[i]}\t {wt[i]} ")

print("\nAvg TAT :", avg_tat)
print("Avg WT :", avg_wt)
```
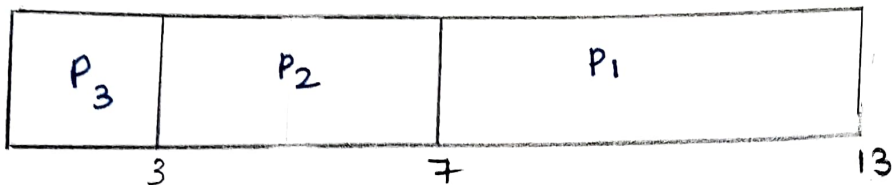
Gantt chart:

P₃	P₂	P₁

3   7   13

Tabulation:

Process	BT (ms)	AT (ms)	CT (ms)	TAT = AT-CT (ms)	WT = TAT-BT (ms)
3	3	0	3	3	0
2	4	0	7	7	3
1	6	0	13	13	7

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

vi sjf.py

python3 sjf.py

Enter the number of processes: 3

Enter PID 1 : 1

Enter PID 2 : 2

Enter PID 3 : 3

Enter BT for PID1 : 6

Enter BT for PID 2 : 4

Enter BT for PID 3 : 3

PID	BT	CT	TAT	WT
3	3	3	3	0
2	4	7	7	3
1	6	13	13	7

Avg. TAT : 7.66667

Avg. WT : 3.33335

Result:

Thus the SJF (shortest Job first) scheduling technique was implemented using Python, executed successfully and gave the expected output.

40