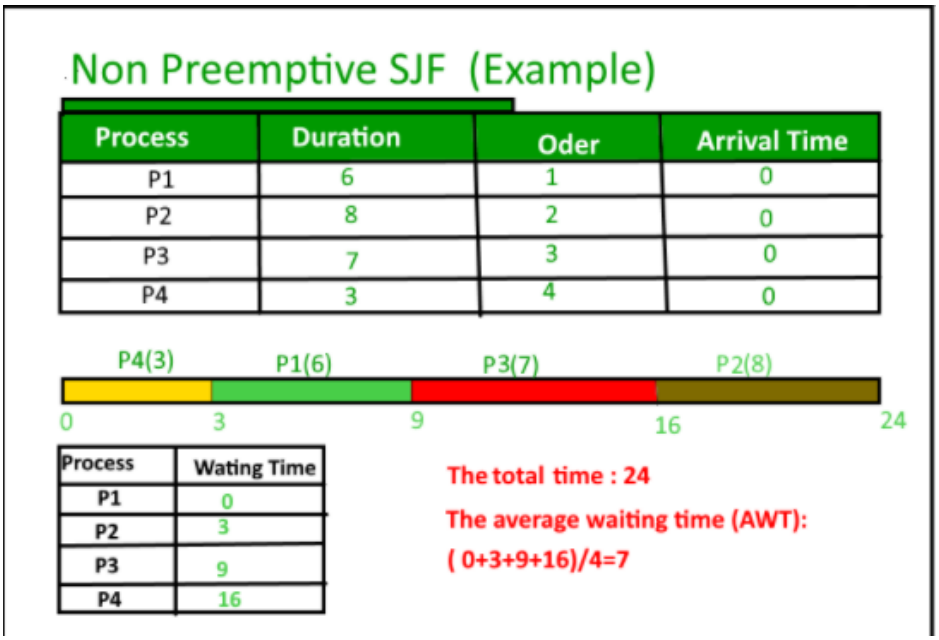
OPERATING SYSTEM LAB 4a

Roll No.: K041	Name: Anish Sudhan Nair
Batch No.: A2/K2	Date: 10/01/2021

Aim: To familiarise and implement the shortest job first scheduling algorithm.

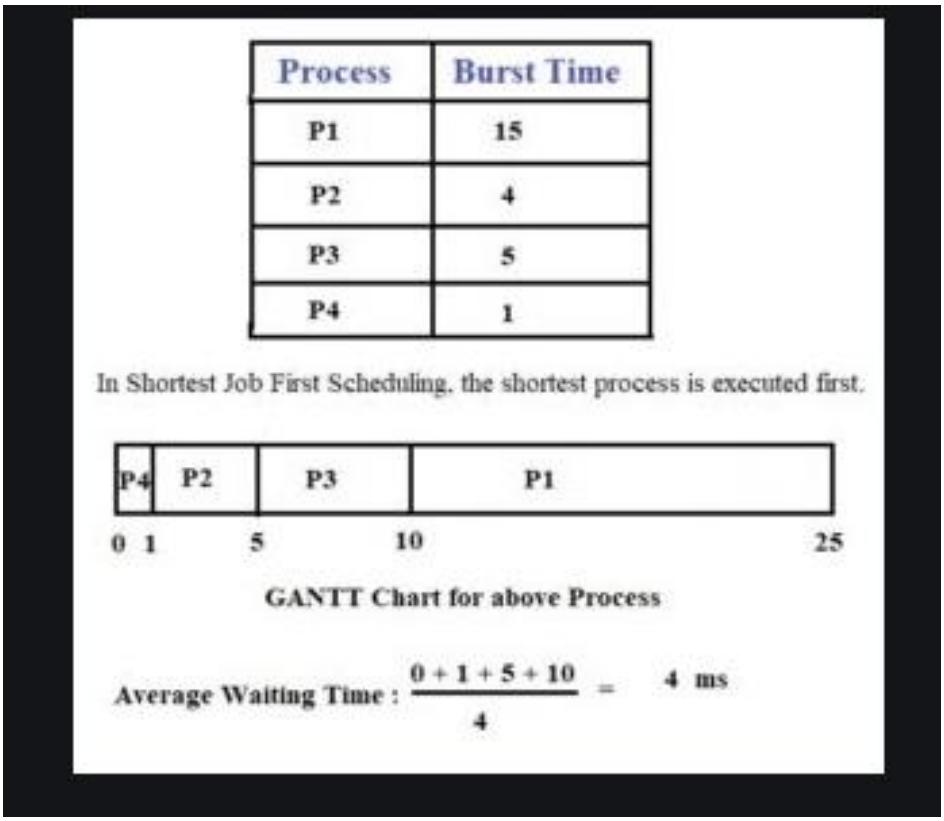
1. Example 1



Output:

```
(base) anish@Anishs-MacBook-Pro Lab % clang sjf_algo.c -o a
(base) anish@Anishs-MacBook-Pro Lab % ./a
Enter the number of processes: 4
Enter the burst time for process 1: 6
Enter the arrival time for process 1: 0
Enter the burst time for process 2: 8
Enter the arrival time for process 2: 0
Enter the burst time for process 3: 7
Enter the arrival time for process 3: 0
Enter the burst time for process 4: 3
Enter the arrival time for process 4: 0


The Average Turn Around Time is: 13.00
The Average Waiting Time is: 7.00%
```

2. Example 2



Process	Burst Time
P1	15
P2	4
P3	5
P4	1

In Shortest Job First Scheduling, the shortest process is executed first.

GANTT Chart for above Process

Average Waiting Time : $\frac{0 + 1 + 5 + 10}{4} = 4$ ms

Output:

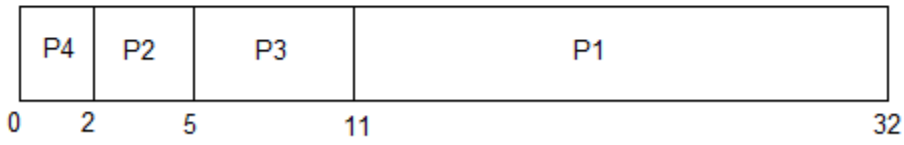
```
(base) anish@Anishs-MacBook-Pro Lab % ./a
Enter the number of processes: 4
Enter the burst time for process 1: 15
Enter the arrival time for process 1: 0
Enter the burst time for process 2: 4
Enter the arrival time for process 2: 0
Enter the burst time for process 3: 5
Enter the arrival time for process 3: 0
Enter the burst time for process 4: 1
Enter the arrival time for process 4: 0


The Average Turn Around Time is: 10.25
The Average Waiting Time is: 4.00%
```

3. Example 3

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

In Shortest Job First Scheduling, the shortest Process is executed first.

Hence the GANTT chart will be following :



Now, the average waiting time will be = (0 + 2 + 5 + 11)/4 = 4.5 ms

Output:

```
(base) anish@Anishs-MacBook-Pro Lab % ./a
Enter the number of processes: 4
Enter the burst time for process 1: 21
Enter the arrival time for process 1: 0
Enter the burst time for process 2: 3
Enter the arrival time for process 2: 0
Enter the burst time for process 3: 6
Enter the arrival time for process 3: 0
Enter the burst time for process 4: 2
Enter the arrival time for process 4: 0


The Average Turn Around Time is: 12.50
The Average Waiting Time is: 4.50%
```

CODE:

```c
#include <stdio.h>
int num_process, processes[10], exit_times[10], arrival_times[10], waiting_times[10];
int burst_times[10], turnAround_times[10];
int temp, temp2;
//shortest job first

int exitTime(int i){
  if ((i==0) || (arrival_times[i]>exit_times[i-1]))
    return (arrival_times[i] + burst_times[i]);
  else
    return (exit_times[i-1] + burst_times[i]);
}

int turnAroundTime(int i){
  return exit_times[i]-arrival_times[i];
}

int waitingTime(int i){
  return turnAround_times[i] - burst_times[i];
}

void avgTime(){
  int totalWaitingTime=0, totalTurnAroundTime=0;
  for (int i = 0; i < num_process; i++) {
    totalWaitingTime+=waiting_times[i];
    totalTurnAroundTime+=turnAround_times[i];
  }
  float avgWaitingTime= (float)totalWaitingTime/(float)num_process;
  float avgTurnAroundTime=(float)totalTurnAroundTime/(float)num_process;

  printf("\n\nThe Average Turn Around Time is: %.2f\n", avgTurnAroundTime );
  printf("The Average Waiting Time is: %.2f",avgWaitingTime );
}

int main(){

  printf("Enter the number of processes: " );
  scanf("%d",&num_process);


  for (int i = 0; i < num_process; i++) {
    printf("Enter the burst time for process %d: ",i+1 );
    scanf("%d", &burst_times[i] );

    printf("Enter the arrival time for process %d: ",i+1 );
    scanf("%d", &arrival_times[i] );

  }

for (int j = 0; j < num_process; j++) {
  for (int i = j; i < num_process; i++) {
    if (burst_times[i]<burst_times[j]) {
```

```
        temp=arrival_times[i];
        arrival_times[i]=arrival_times[j];
        arrival_times[j]=temp;

        temp2=burst_times[i];
        burst_times[i]=burst_times[j];
        burst_times[j]=temp2;
      }
    }
  }

    for (int i = 0; i < num_process; i++) {

      exit_times[i]=exitTime(i);

      turnAround_times[i]=turnAroundTime(i);

      waiting_times[i]=waitingTime(i);
    }

    avgTime();

    return 0;
  }

```

CONCLUSION:

In this lab, we were to implement and demonstrate the working of the shortest job first algorithm. By actually coding the algorithm, it helped to reinforce the working of this process and the manner in which it schedules the processes in a CPU.