Lab-4.

Title:- First Come First Serve Scheduling Algorithm

For Non-Preemptive:-

Algorithm

1) Print Input no of processes
2) Scan & put in variable n;
3) Input burst time; //maintain an array for process & burst time
4) waiting time for process $P_0 = 0$
5) turnaround time = burst time [n] + waiting time [n]
6) waiting time = burst time [n-1] + waiting time [n-1]
7) print the output with average value.

Source Code:-

```
#include<stdio.h>
int main(){
        int n, i;
        printf("Enter the number of processes:");
        scanf("%d", &n);
        int burst_time[n], waiting_time[n], turnaround_time[n];
```

```
float total_waiting_time=0, total_turnaround_
        time= 0;

for(i=0; i<n; i++){
    printf("Enter the burst time of Process
            %d : ", i+1);
    scanf("%d", &burst_time[i]);

    }
waiting_time[0]=0;
for(i=1; i<n; i++){
    waiting_time[i]=waiting_time[i-1]+burst_
                    time[i-1];
    total_waiting_time += waiting_time[i];
    }
for (i=0; i<n; i++){
  turnaround_time[i]= waiting_time[i] +
                burst_time[i];
  total_turnaround_time= turnaround_time[i];
    }
printf ("Process\t Burst Time\t Waiting
        Time\t Turnaround Time\n");
for(i=0; i<n; i++){
   printf("%d\t%d\t%d\t%d\t", i+1, burst_
   time[i], waiting_time[i], turnaround_tim-
   e[i]);
}
```

```
printf("Average Waiting Time: %.2f\n", total_
        waiting_time/n);
printf("Average Turnaround Time: %.2f\n",
        total_turnaround_time/n);

return 0;
}
```

Conclusion:-
In this lab, we get familiar First come First serve scheduling (non-preemptive) algorithm.

2080-09-23

Title:- Implementation of Shortest Job First Algorithm

Theory:-

Shortest Job First (SJF):

- Allocate the CPU to the process with least CPU burst time.
- If two processes have same CPU burst, then FCFS is used to break the tie.
- Two schemes:

• Non-preemptive
  - once CPU given to the process it cannot be preempted until completes its CPU burst.

• Preemptive (also called shortest Remaining First SRJF)

- ~~Preemptive~~ If a new Process arives with CPU burst length less than remaining time of current executing process, preempt.

SOURCE CODE:

```
#include<stdio.h>
int main(){
    int n,i,j,temp,sum=0;
    float avg_wt,avg_tat,twt,ttat;
    int bt[20],wt[20],tat[20],p[20];
    printf("Enter the number of process:");
    scanf("%d",&n);
```

```
for(i=0;i<n;i++){
    printf("Enter the burst time
            of process %d:",i+1);
    scanf("%d",&bt[i+1]);
    p[i]=i+1; //initialize process
                array in original
                order
{
//sort the burst time in ass
 cending order.
for(i=0;i<n;i++){
if(bt[i]>bt[j])
  { temp=bt[i];
    bt[i]=bt[j];
    bt[j]= temp;
   //sort process array
   along with burst time arrays
    temp=p[i];
    p[i]=p[j];
    p[j]=temp;
  }
}
}
//calculate waiting time & turnaround
times
    wt[0]=0;
    tat[0]=bt[0];
    for(i=1;i<n;i++){
```

```
    wt[i] = wt[i-1] + bt[i-1];
    twt += wt[i];
    tat[i] = bt[i-1] + bt[i];
    ttat += tat[i];
}
// print the results
printf("process Burst time waiting time Turnaround time\n");
for(i=0; i<n; i++){
    printf("%d\t%d\t%d\t%d\t",
        p[i], bt[i], wt[i], tat[i]);
}
printf("Average waiting time:%
    0.2f\n", twt\n);
printf("Average turnaround time:%
    0.2f\n", (ttat + tat[0])\n);
return 0;
}
```

DISCUSSION: In this lab we discussed about the shortest Job First algorithm, we got to know about it with the example and also by implementing it in a source code / program. we do a program.

CONCLUSION: