**Practical Assignment-5**

//Name:- Rohit Lad

Roll:- TIA29

//Batch:- B

**//Program:-**

#include <stdio.h>

#define MAX\_PROCESSE 10

int main()

{

int n, i;

int brust\_time[MAX\_PROCESSE], remaining\_time[MAX\_PROCESSE];

int process\_order[MAX\_PROCESSE], waiting\_time[MAX\_PROCESSE], turnaround\_time[MAX\_PROCESSE];

int arrrival\_time[MAX\_PROCESSE];

int current\_time = 0;

int complete = 0;

printf("Enter the number of processes: ");

scanf("%d", &n);

printf("Enter the burst time for each process:\n");

for (i = 0; i < n; i++)

{

printf("Burst time for process %d: ", i + 1);

scanf("%d", &brust\_time[i]);

remaining\_time[i] = brust\_time[i];

process\_order[i] = i + 1;

}

printf("Enter arrival times for each process:\n");

for (i = 0; i < n; i++)

{

printf("Arrival time for process %d: ", i + 1);

scanf("%d", &arrrival\_time[i]);

}

while (complete < n)

{

int shortest\_job = -1;

int shortest\_time = 9999;

for (i = 0; i < n; i++)

{

if (arrrival\_time[i] <= current\_time && remaining\_time[i] > 0 && remaining\_time[i] < shortest\_time)

{

shortest\_job = i;

shortest\_time = remaining\_time[i];

}

}

if (shortest\_job == -1)

{

current\_time++;

}

else

{

remaining\_time[shortest\_job]--;

current\_time++;

if (remaining\_time[shortest\_job] == 0)

{

complete++;

waiting\_time[shortest\_job] = current\_time - arrrival\_time[shortest\_job] - brust\_time[shortest\_job];

if (waiting\_time[shortest\_job] < 0)

{

waiting\_time[shortest\_job] = 0;

}

turnaround\_time[shortest\_job] = waiting\_time[shortest\_job] + brust\_time[shortest\_job];

}

}

}

double avg\_waiting\_time = 0, avg\_turnaround\_time = 0;

for (i = 0; i < n; i++)

{

avg\_waiting\_time += waiting\_time[i];

avg\_turnaround\_time += turnaround\_time[i];

}

avg\_waiting\_time /= n;

avg\_turnaround\_time /= n;

printf("\n\nProcess\tBurst Time\tArrival Time\tWaiting Time\tTurnaround Time\n");

for (i = 0; i < n; i++)

{

printf("P%d\t%d\t\t%d\t\t%d\t\t%d\n", process\_order[i], brust\_time[i],

arrrival\_time[i], waiting\_time[i], turnaround\_time[i]);

}

printf("\nAverage Waiting Time: %f", avg\_waiting\_time);

printf("\nAverage Turnaround Time: %f", avg\_turnaround\_time);

return 0;

}

**Output:-**

student@student:~$ gcc TIA\_29OS\_5.c

student@student:~$ ./a.out

Enter the number of processes: 5

Enter the burst time for each process:

Burst time for process 1: 2

Burst time for process 2: 4

Burst time for process 3: 6

Burst time for process 4: 3

Burst time for process 5: 8

Enter arrival times for each process:

Arrival time for process 1: 0

Arrival time for process 2: 3

Arrival time for process 3: 2

Arrival time for process 4: 1

Arrival time for process 5: 4

Process Burst Time Arrival Time Waiting Time Turnaround Time

P1 2 0 0 2

P2 4 3 2 6

P3 6 2 7 13

P4 3 1 1 4

P5 8 4 11 19

Average Waiting Time: 4.200000