**4,17**

**Implement the C program for CPU Scheduling Algorithms: Shortest Job First (Pre-emptive) and Round Robin with different arrival time.**

#include <stdio.h>

int main()

{

int arrival\_time[10], burst\_time[10], rt[10];

int n, i, smallest, count = 0, time;

double wait\_time = 0, turnaround\_time = 0, completiontime;

float average\_waiting\_time = 0, average\_turnaround\_time = 0;

printf("\nEnter the Total Number of Processes:");

scanf("%d", &n);

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

{

printf("Enter Arrival time and Burst time of Process :%d\n", i + 1);

printf("Enter Arrival Time:");

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

printf("Enter Burst Time:");

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

rt[i] = burst\_time[i];

}

burst\_time[9] = 9999;

for (time = 0; count != n; time++)

{

smallest = 9;

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

{

if (arrival\_time[i] <= time && burst\_time[i] < burst\_time[smallest] && burst\_time[i] > 0)

{

smallest = i;

}

}

burst\_time[smallest]--;

if (burst\_time[smallest] == 0)

{

count++;

completiontime= time + 1;

wait\_time = wait\_time + completiontime - arrival\_time[smallest] - rt[smallest];

turnaround\_time = turnaround\_time + completiontime - arrival\_time[smallest];

}

}

average\_waiting\_time = wait\_time / n;

average\_turnaround\_time = turnaround\_time / n;

printf("Average Waiting Time:%lf\n", average\_waiting\_time);

printf("Average Turnaround Time:%lf", average\_turnaround\_time);

return 0;

}

Rr

#include <stdio.h>

int main()

{

int i, n, time = 0, count = 0, y, qt, wt = 0, tat = 0, at[10], bt[10], rt[10];

float avg\_wt, avg\_tat;

printf(" Total number of process in the system: ");

scanf("%d", &n);

y = n; // Assign the number of process to variable y

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

{

printf("\n Enter the Arrival and Burst time of the Process[%d]\n", i + 1);

printf(" Arrival time is: "); // Accept arrival time

scanf("%d", &at[i]);

printf("Burst time is: "); // Accept the Burst time

scanf("%d", &bt[i]);

rt[i] = bt[i]; // store the burst time in temp array

}

printf("Enter the Time Quantum for the process: \t");

scanf("%d", &qt);

printf("\n Process No \t\t Burst Time \t\t TAT \t\t Waiting Time ");

for (time = 0, i = 0; y != 0;)

{

if (rt[i] <= qt && rt[i] > 0) // define the conditions

{

time = time + rt[i];

rt[i] = 0;

count = 1;

}

else if (rt[i] > 0)

{

rt[i] = rt[i] - qt;

time = time + qt;

}

if (rt[i] == 0 && count == 1)

{

y--; // decrement the process no.

printf("\nProcess No :%d \t\t %d\t\t\t\t %d\t\t\t %d", i + 1, bt[i], time - at[i], time - at[i] - bt[i]);

wt = wt + time - at[i] - bt[i];

tat = tat + time - at[i];

count = 0;

}

if (i == n - 1)

{

i = 0;

}

else if (at[i + 1] <= time)

{

i++;

}

else

{

i = 0;

}

}

// represents the average waiting time and Turn Around time

avg\_wt = wt \* 1.0 / n;

avg\_tat = tat \* 1.0 / n;

printf("\n Average Turn Around Time: \t%f", avg\_wt);

printf("\n Average Waiting Time: \t%f", avg\_tat);

return 0;

}