**Lab Program Number: 3 Date: 2079/03/**

**Title: Write a program to implement Round Robin Scheduling algorithm(RR).**

**Round Robin:**

Round Robin is the preemptive process scheduling algorithm. Here, each process is provided the fixed time to execute which is called quantum time. Once a process is executed for a given period of time then it is preempted and the other process executes for a given period of time.

**Source Code:**

|  |
| --- |
| *//program for round robin process scheduling*  #include<stdio.h>  #include<conio.h>  int main()  {  *// initlialize of variable*  int i, n, sum=0,count=0, y, quantum, wt=0, tat=0, at[10], bt[10], temp[10];  float avg\_wt, avg\_tat;  printf(" enter the total number of process: ");  scanf("%d", &n);  y = n; *// Assign the number of process to variable y*    *// entering the details of the process like Arrival time and the Burst Time*  *//using the for loop*  for(i=0; i<n; i++)  {  printf("\n the Arrival and Burst time of the Process[%d]\n", i+1);  printf(" Arrival time=\t");  scanf("%d", &at[i]);  printf(" Burst time=\t");  scanf("%d", &bt[i]);  temp[i] = bt[i]; *// temp array for storing the burst time*  }  *//time quantum acceptance*  printf("Enter the Time Quantum=\t");  scanf("%d", &quantum);  *// Display the process No, burst time, Turn Around Time and the waiting time*  printf("\n Process No \t\t\t Burst Time \t\t\t TAT \t\t\t Waiting Time ");  for(sum=0, i = 0; y!=0; )  {  if(temp[i] <= quantum && temp[i] > 0)  {  sum = sum + temp[i];  temp[i] = 0;  count=1;  }  else if(temp[i] > 0)  {  temp[i] = temp[i] - quantum;  sum = sum + quantum;  }  if(temp[i]==0 && count==1)  {  y--;  printf("\n%d\t\t\t\t %d\t\t\t\t %d\t\t\t %d", i+1, bt[i], sum-at[i], sum-at[i]-bt[i]);  wt = wt+sum-at[i]-bt[i];  tat = tat+sum-at[i];  count =0;  }  if(i==n-1)  {  i=0;  }  else if(at[i+1]<=sum)  {  i++;  }  else  {  i=0;  }  }  *// now gives the average waiting and turnaroud time as*  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\t%f", avg\_tat);  getch();  } |

IDE: Dev-C++

Programming Language: C-programming

**Output:**

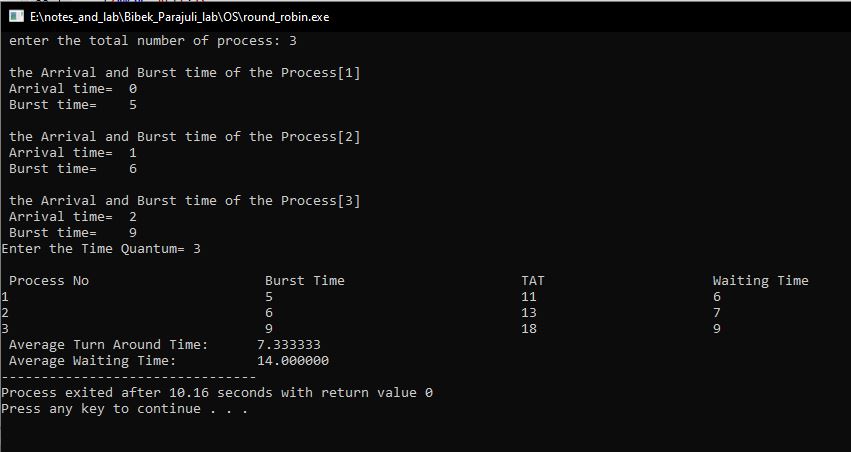


Fig: RR Output