**Lab Program Number: 2 Date: 2079/03/21**

**Title: Write a program to implement Shortest Job First (SJF)**

**Shortest Job First (SJF)**

**Shortest Job First (SJF)** is a process scheduling algorithm in which the process having the smallest execution time is chosen for the next execution. This scheduling method is preemptive that is when any of the process enters into critical section than it must complete their job.

**Source Code:**

|  |
| --- |
| // Write a program to implement Shortest Job First (SJF)  #include<stdio.h>  int main()  {  int bt[10],p[10],wt[10],tat[10],i,j,n,pos,temp;  float avgWt,avgTat,total=0;  printf("Enter number of process:");  scanf("%d",&n);    printf("\nEnter Burst Time:\n");  for(i=0;i<n;i++)  {  printf("p%d:",i+1);  scanf("%d",&bt[i]);  p[i]=i+1;  }    for(i=0;i<n;i++)  { //sorting of the brust time using bubble sort  pos=i;  for(j=i+1;j<n;j++)  {  if(bt[j]<bt[pos])  pos=j;  }    temp=bt[i];  bt[i]=bt[pos];  bt[pos]=temp;    temp=p[i];  p[i]=p[pos];  p[pos]=temp;  }    wt[0]=0; // initially waiting time is 0.  for(i=1;i<n;i++)  {  wt[i]=0;  for(j=0;j<i;j++)  wt[i]+=bt[j];    total+=wt[i];  }  avgWt=total/n;  total=0;    printf("\nProcess\t Burst Time \tTurnAround Time \tWaiting Time");  for(i=0;i<n;i++)  {  tat[i]=bt[i]+wt[i];  total+=tat[i];  printf("\np%d\t\t %d\t\t %d\t\t\t%d",p[i],bt[i],tat[i],wt[i]);  }    avgTat=total/n;    printf("\nAverage Turnaround Time=%f",avgTat);  printf("\n\nAverage Waiting Time=%f",avgWt);  } |

IDE: Dev-C++

Programming Language: C-programming

**Output:**

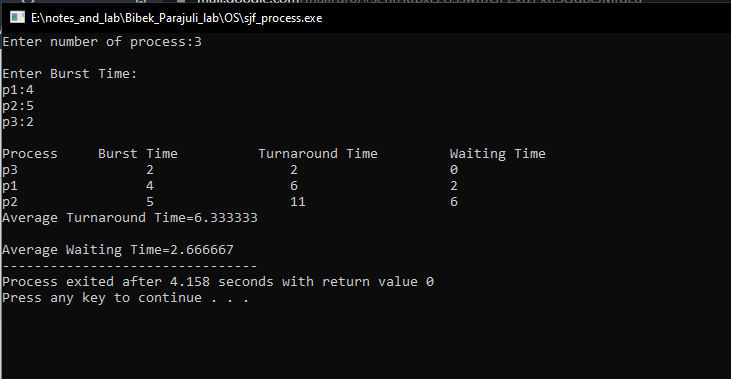


Fig: SJF Output