Saurabh Kumar Singh(16100051)

SRTF Scheduling

ALGORITHM::
Mode:- Preemptive
Criteria:- Burst Time

1- Traverse until all process gets completely executed.

a) Find process with minimum remaining time at every single time lap.
b) Reduce its time by 1.
c) Check if its remaining time becomes 0
d) Increment the counter of process completion.
e) Completion time of current process = current_time +1;
e) Calculate waiting time for each completed process.
wt[i]= Completion time - arrival_time-burst_time
f)Increment time lap by one.

2- Find turnaround time (waiting_time+burst_time).

CODE::

```
#include<iostream>
using namespace std;
int main()
{
      int
n,i,v[10]={0},j,temp=0,to=0,ti=0,tem=0,te=0,k,xb[10],xa[10],a[10],b[10],t,coun
t=0,pid[10],c;
      int ct[10],wt[10],tat[10];
      double tat_av=0,ct_av=0,wt_av=0;
      cout<<"Enter the number of process"<<"\n";
      cin>>n;
      cout<<"Enter the arrival time "<<"\n";
      for(i=0;i<n;i++)
```

```
{
           //scanf("%d",&pid[i]);
      pid[i]=i;
      cin>>a[i];
}
cout<<"enter the burst time "<<"\n";
for(i=0;i<n;i++)
      cin>>b[i];
for(i=0;i<n;i++)
{
      xb[i]=b[i];
      xa[i]=a[i];

}
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
      if(a[i]>a[j])
      {
      temp=a[i];
      a[i]=a[j];
      a[j]=temp;


      ti=xa[i];
      xa[i]=xa[j];
      xa[j]=ti;


      tem=pid[i];
      pid[i]=pid[j];
      pid[j]=tem;



      te=b[i];
      b[i]=b[j];
      b[j]=te;

      to=xb[i];
      xb[i]=xb[j];
      xb[j]=to;
      }

}
}
```

```
          for(j=0;j<n;j++)
          {
               cout<<"The order of process is as follows:"<<"\n";
               cout<<"Process = "<<pid[j] <<": arrival time = "<<a[j]<<" :
burst time= "<<b[j]<<"\n";

     }

count=0;
c=0;

     for(t=0;;t++)
     {


               for(j=0;j<n;j++)
               {
                    if((t>=a[j])&&(xb[count]>xb[j])&&(v[j]!=1))
                    {
                    count=j;
                    xb[count]=xb[count]-1;
                    goto Label;
                    }
               }

xb[count]=xb[count]-1;
Label:
cout<<"\n"<<"For time "<< t<<"to"<<(t+1)<<" : process "<<pid[count]<<"\n";
               for(k=0;k<n;k++)
               {
                    if((xb[k]==0)&&(v[k]!=1))
                    {


                         c++;

                         v[k]=1;
                         xb[k]=999;
                         ct[k]=t+1;

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

                              for(j=0;j<n;j++)
                              {
                              if((xb[count]>xb[j])&&(a[j]<t))
                                   count=j;
                              }
```

```
                               }
                         }

                  }
if(c==n)
break;

      }


                  for(k=0;k<n;k++)
                  {
                        tat[k]=ct[k]-a[k];
                        wt[k]=tat[k]-b[k];
            }


                  for(k=0;k<n;k++)
                  {

                    cout<<"ct["<<k<<"] is"<<ct[k]<<"\n";
                  }
                  for(k=0;k<n;k++)
                  {
                        cout<<"wt["<<k<<"] is"<<wt[k]<<"\n";


                  }
                  for(k=0;k<n;k++)
                  {
                        cout<<"tat["<<k<<"] is"<<tat[k]<<"\n";


                  }



                  for(k=0;k<n;k++)
                  {
                        tat_av=tat_av+tat[k];
                        wt_av=wt_av+wt[k];

            }

        cout<<"Average turnaround time  is"<<tat_av/n<<"\n";
        cout<<"Average waiting  time  is"<<wt_av/n<<"\n";

}
```

INPUT::

```
Enter the number of process
4
Enter the arrival time
0 2 4 7
enter the burst time
7 4 2 1
```

OUTPUT::

```
The order of process is as follows:
Process = 0: arrival time = 0 : burst time= 7
The order of process is as follows:
Process = 1: arrival time = 2 : burst time= 4
The order of process is as follows:
Process = 2: arrival time = 4 : burst time= 2
The order of process is as follows:
Process = 3: arrival time = 7 : burst time= 1

For time 0to1 : process 0

For time 1to2 : process 0

For time 2to3 : process 1

For time 3to4 : process 1

For time 4to5 : process 1

For time 5to6 : process 1

For time 6to7 : process 2

For time 7to8 : process 2

For time 8to9 : process 3

For time 9to10 : process 0

For time 10to11 : process 0

For time 11to12 : process 0

For time 12to13 : process 0

For time 13to14 : process 0
```

```
ct[0] is14
ct[1] is6
ct[2] is8
ct[3] is9
wt[0] is7
wt[1] is0
wt[2] is2
wt[3] is1
tat[0] is14
tat[1] is4
tat[2] is4
tat[3] is2
Average turnaround time  is6
Average waiting  time  is2.5
```