Assignment No: 2

Name: Kedar Kale

Roll No: CETTY129

Subject: OS Practical

Topic: CPU Scheduling Algorithm

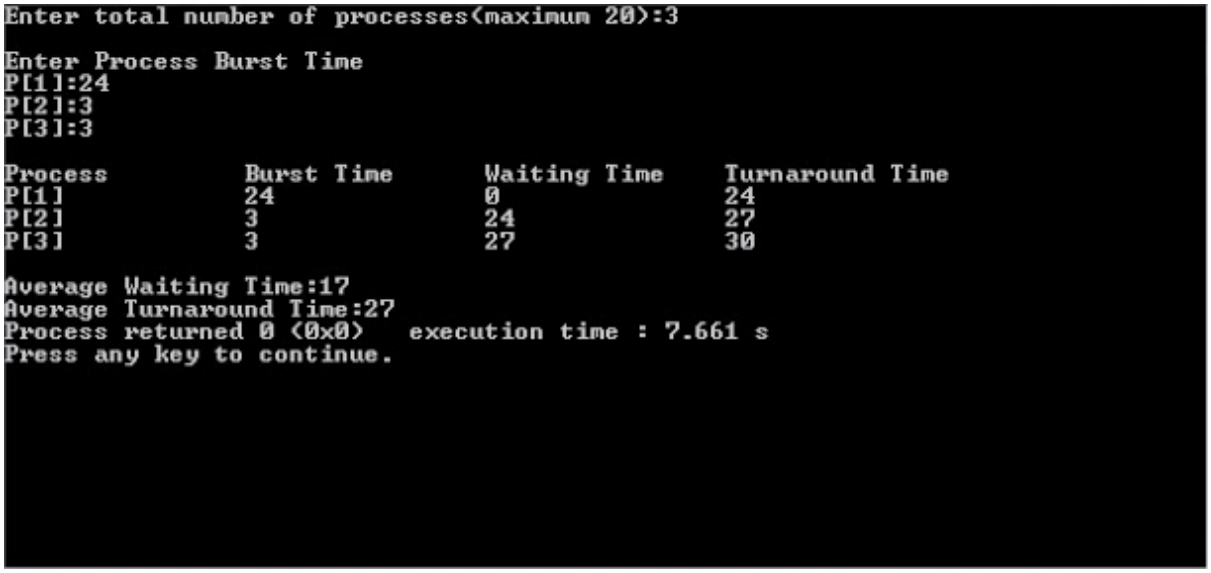
First Come First Serve Algorithm

Code:

```
#include<iostream>

using namespace std;

int main()
{
   int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;
   cout<<"Enter total number of processes(maximum 20):";
   cin>>n;

   cout<<"\nEnter Process Burst Time\n";
   for(i=0;i<n;i++)
   {
      cout<<"P["<<i+1<<"]:";
      cin>>bt[i];
   }

   wt[0]=0;    //waiting time for first process is 0

   //calculating waiting time
   for(i=1;i<n;i++)
   {
      wt[i]=0;
      for(j=0;j<i;j++)
         wt[i]+=bt[j];
   }

   cout<<"\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time";

   //calculating turnaround time
   for(i=0;i<n;i++)
   {
      tat[i]=bt[i]+wt[i];
      avwt+=wt[i];
      avtat+=tat[i];
      cout<<"\nP["<<i+1<<"]"<<"\t\t"<<bt[i]<<"\t\t"<<wt[i]<<"\t\t"<<tat[i];
   }
```

```
    avwt/=i;
    avtat/=i;
    cout<<"\n\nAverage Waiting Time:"<<avwt;
    cout<<"\nAverage Turnaround Time:"<<avtat;

    return 0;
}
```

Output:



Shortest Job First Algorithm

Code:

```
#include<iostream>

using namespace std;

int main()
{
    int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
    float avg_wt,avg_tat;
    cout<<"Enter number of process:";
    cin>>n;

    cout<<"\nEnter Burst Time:\n";
    for(i=0;i<n;i++)
    {
        cout<<"p"<<i+1<<":";
        cin>>bt[i];
        p[i]=i+1;           //contains process number
    }

    //sorting burst time in ascending order using selection sort
    for(i=0;i<n;i++)
    {
        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;            //waiting time for first process will be zero

    //calculate waiting time
    for(i=1;i<n;i++)
    {
        wt[i]=0;
        for(j=0;j<i;j++)
            wt[i]+=bt[j];

        total+=wt[i];
    }

    avg_wt=(float)total/n;      //average waiting time
    total=0;

    cout<<"\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time";
    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];     //calculate turnaround time
        total+=tat[i];
        cout<<"\n"<<p[i]<<"\t\t"<<bt[i]<<"\t\t"<<wt[i]<<"\t\t"<<tat[i];
    }

    avg_tat=(float)total/n;     //average turnaround time
    cout<<"\n\nAverage Waiting Time=%f"<<avg_wt;
    cout<<"\nAverage Turnaround Time=%f\n"<<avg_tat;
    return 0;
}
```

Output:

Round Robbin Scheduling Algorithm:

Code:

```
#include<stdio.h>

int main()
{

  int count,j,n,time,remain,flag=0,time_quantum;
  int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
  printf("Enter Total Process:\t ");
  scanf("%d",&n);
  remain=n;
  for(count=0;count<n;count++)
  {
    printf("Enter Arrival Time and Burst Time for Process Process Number %d :",count+1);
    scanf("%d",&at[count]);
    scanf("%d",&bt[count]);
    rt[count]=bt[count];
  }
  printf("Enter Time Quantum:\t");
  scanf("%d",&time_quantum);
  printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");
  for(time=0,count=0;remain!=0;)
  {
    if(rt[count]<=time_quantum && rt[count]>0)
    {
      time+=rt[count];
      rt[count]=0;
      flag=1;
    }
    else if(rt[count]>0)
    {
      rt[count]-=time_quantum;
      time+=time_quantum;
    }
    if(rt[count]==0 && flag==1)
    {
      remain--;
      printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-at[count]-bt[count]);
      wait_time+=time-at[count]-bt[count];
      turnaround_time+=time-at[count];
      flag=0;
    }
    if(count==n-1)
      count=0;
    else if(at[count+1]<=time)
      count++;
    else
      count=0;
  }
  printf("\nAverage Waiting Time= %f\n",wait_time*1.0/n);
```

```
  printf("Avg Turnaround Time = %f",turnaround_time*1.0/n);

  return 0;
}
```

Output:

```
Average Waiting Time= 5.250000
Avg Turnaround Time = 9.500000tusharsoni@tusharsoni-Lenovo-G50-70:~/Desktop$ ./
.out
Enter Total Process:     4
Enter Arrival Time and Burst Time for Process Process Number 1 :0
9
Enter Arrival Time and Burst Time for Process Process Number 2 :1
5
Enter Arrival Time and Burst Time for Process Process Number 3 :2
3
Enter Arrival Time and Burst Time for Process Process Number 4 :3
4
Enter Time Quantum:     5


Process |Turnaround time|waiting time

P[2]    |       9       |       4
P[3]    |       11      |       8
P[4]    |       14      |       10
P[1]    |       21      |       12

Average Waiting Time= 8.500000
Avg Turnaround Time = 13.750000tusharsoni@tusharsoni-Lenovo-G50-70:~/Desktop$
```

Priority Scheduling Algorithm:

Code:

```
#include<iostream>

using namespace std;

int main()
{
   int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg_wt,avg_tat;
   cout<<"Enter Total Number of Process:";
   cin>>n;

   cout<<"\nEnter Burst Time and Priority\n";
   for(i=0;i<n;i++)
   {
      cout<<"\nP["<<i+1<<"]\n";
      cout<<"Burst Time:";
      cin>>bt[i];
      cout<<"Priority:";
      cin>>pr[i];
```

```
   p[i]=i+1;          //contains process number
}

//sorting burst time, priority and process number in ascending order using selection sort
for(i=0;i<n;i++)
{
   pos=i;
   for(j=i+1;j<n;j++)
   {
      if(pr[j]<pr[pos])
         pos=j;
   }

   temp=pr[i];
   pr[i]=pr[pos];
   pr[pos]=temp;

   temp=bt[i];
   bt[i]=bt[pos];
   bt[pos]=temp;

   temp=p[i];
   p[i]=p[pos];
   p[pos]=temp;
}

wt[0]=0;           //waiting time for first process is zero

//calculate waiting time
for(i=1;i<n;i++)
{
   wt[i]=0;
   for(j=0;j<i;j++)
      wt[i]+=bt[j];

   total+=wt[i];
}

avg_wt=total/n;      //average waiting time
total=0;

cout<<"\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time";
for(i=0;i<n;i++)
{
   tat[i]=bt[i]+wt[i];     //calculate turnaround time
   total+=tat[i];
   cout<<"\nP["<<p[i]<<"]\t\t  "<<bt[i]<<"\t\t    "<<wt[i]<<"\t\t\t"<<tat[i];
}

avg_tat=total/n;     //average turnaround time
cout<<"\n\nAverage Waiting Time="<<avg_wt;
cout<<"\nAverage Turnaround Time="<<avg_tat;
```

```
    return 0;
}
```

Output: