Rajshahi University of Engineering and Technology

Department of Computer Science and Engineering

Assignment on different scheduling algorithms

Course No: CSE 3201

Submitted to:

Firoz Mahmud

Assistant Professor

Dept. of CSE, RUET

Submitted by:

Sabit Ahmed

Roll: 1503056

Class: 3rd year, Even semester

First-Come, First-Served Scheduling:

Code:

```cpp
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n,m,arr1[10]={},arr2[10]={},arr3[10]={},arr4[10]={},sum=0,sum2=0,sum3=0;
    cout<<"Enter number of process"<<endl;
    cin>>n;
    cout<<"Enter burst time"<<endl;
    for(int i=1;i<=n;i++){
        cin>>arr1[i];
    }
    m=n+1;

    cout<<endl<<endl;
    for(int i=0;i<m;i++){
        sum=sum+arr1[i];//gantt-chart
        arr2[i]=sum;
        cout<<arr2[i]<<" ";
    }
    arr1[0]=arr2[1];
    for(int i=0;i<n;i++){
```

```cpp
        sum3=sum3+arr2[i+1];

        arr1[i]=sum3;

        sum2=sum2+arr2[i];

        arr3[i]=sum2;

        }
    /*for(int i=0;i<m;i++){

        sum3=sum3+arr2[i+1];

        arr1[i]=sum3;

    }*/

    cout<<endl<<endl;

    cout<<"Avg. waiting time: "<<float(sum2)/n<<endl;

    cout<<"Avg. turnaround time: "<<float(sum3)/n<<endl;

}
```

Screenshot:

Shortest-Job-First Scheduling(Non-Preemptive):

Code:

```cpp
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n,m,arr1[10]={},arr2[10]={},arr3[10]={},arr4[10]={},sum=0,sum2=0,sum3=0;
    cout<<"Enter number of process"<<endl;
    cin>>n;
    cout<<"Enter burst time"<<endl;
    for(int i=1;i<=n;i++){
        cin>>arr1[i];
    }
    m=n+1;
    sort(arr1,arr1+m);


    cout<<endl<<endl;
    for(int i=0;i<m;i++){
        sum=sum+arr1[i];//gantt-chart
        arr2[i]=sum;
        cout<<arr2[i]<<" ";
    }
```
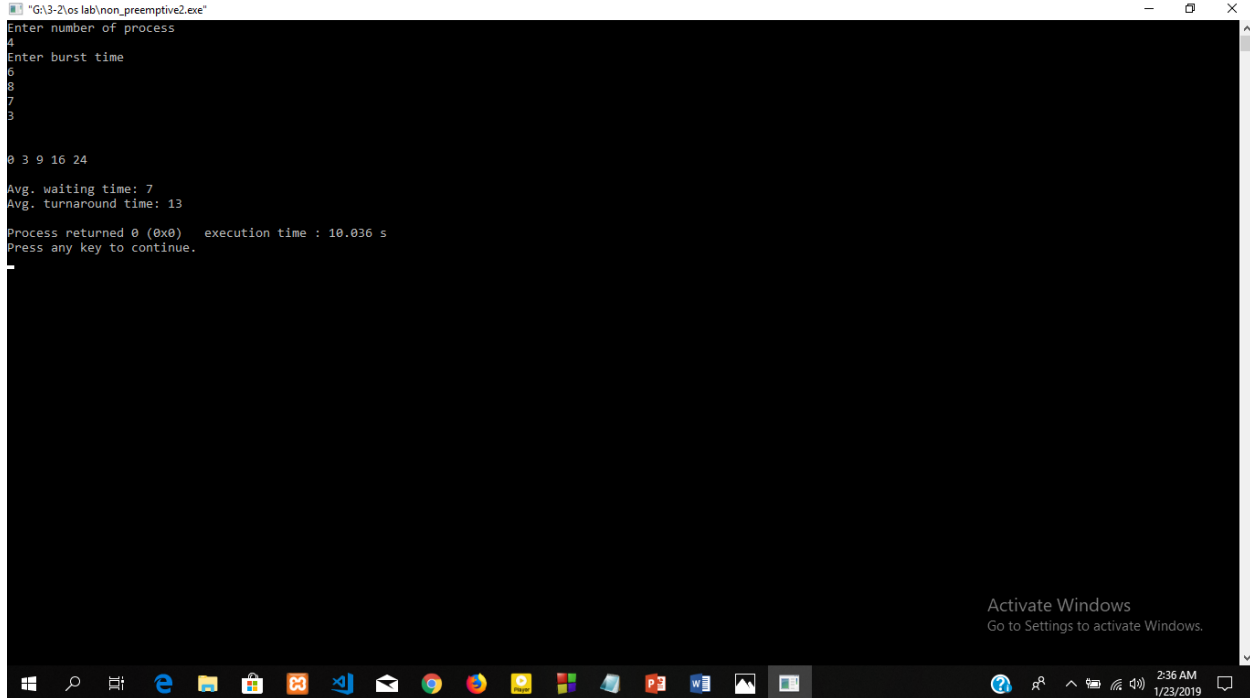
```cpp
    arr1[0]=arr2[1];
    for(int i=0;i<n;i++){
        sum3=sum3+arr2[i+1];
        arr1[i]=sum3;
        sum2=sum2+arr2[i];
        arr3[i]=sum2;
    }
    /*for(int i=0;i<m;i++){
        sum3=sum3+arr2[i+1];
        arr1[i]=sum3;
    }*/
    cout<<endl<<endl;
    cout<<"Avg. waiting time: "<<float(sum2)/n<<endl;
    cout<<"Avg. turnaround time: "<<float(sum3)/n<<endl;
}
```

Screenshot:

Priority Scheduling(Non-Preemptive):

Code:

#include<bits/stdc++.h>

using namespace std;

struct object{

    char process[10];

    int burst_time;

    int priority;

};

```cpp
bool comp(object a,object b){
    return a.priority<b.priority;
}


int main()
{
    object obj[100];
    int n,sum=0,tot_t=0,wait_t=0,g_c[100],t_t[100],w_t[100];
    cout<<"Enter no. of processes:"<<endl;
    cin>>n;
    cout<<"Enter process name, burst time and priority:"<<endl;
    for(int i=0;i<n;i++){
        cin>>obj[i].process>>obj[i].burst_time>>obj[i].priority;
    }
    std::sort(obj,obj+n,comp);
    cout<<endl<<endl;
    for(int i=0;i<n;i++){
        cout<<obj[i].process<<" "<<obj[i].burst_time<<" "<<obj[i].priority<<endl;
    }
    cout<<endl<<endl;
    g_c[0]=0;
    //cout<<g_c[0]<<"  ";
    for(int i=0;i<=n;i++){
        sum=sum+obj[i].burst_time;
```

```
            g_c[i+1]=sum;

            cout<<g_c[i]<<"   ";

        }


        w_t[0]=0;

        t_t[0]=obj[0].burst_time;

        for(int i=1;i<n;i++){

            t_t[i]=t_t[i-1]+obj[i].burst_time;

            w_t[i]=t_t[i-1];

        }

        cout<<endl<<endl;

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

            tot_t=tot_t+t_t[i];

            wait_t=wait_t+w_t[i];

            cout<<w_t[i]<<"   "<<t_t[i]<<endl;

        }

        cout<<endl<<endl;

        cout<<"Avg. waiting time: "<<float(wait_t)/n<<endl;

        cout<<"Avg. turnaround time: "<<float(tot_t)/n<<endl;

    }
```

Screenshot:

Round-Robin:

Code:

```cpp
#include<bits/stdc++.h>
using namespace std;

struct object{
    char process[10];
    int burst_time;
    double l;
};
int main()
{
```

```cpp
object obj[100];
int
n,a,count_g=0,count_l=0,q_t,sum=0,tot_t=0,wait_t=0,g_c[100],t_t[100],w_t[100];
double l;
cout<<"Enter no. of processes:"<<endl;
cin>>n;
cout<<"Enter quantum time:"<<endl;
cin>>q_t;
cout<<"Enter process name and burst time:"<<endl;
for(int i=0;i<n;i++){
    cin>>obj[i].process>>obj[i].burst_time;
    if(q_t<obj[i].burst_time){
        l=double(obj[i].burst_time)/q_t;
        obj[i].l=ceil(l);
        count_g=count_g+obj[i].l;
    }
    else{
        count_l++;
    }
}
//cout<<obj[0].l;
cout<<endl<<endl;
g_c[0]=0;
//cout<<g_c[0]<<"  ";
n=count_g+count_l;
```

```cpp
//cout<<n;
for(int i=0;i<=n;i++){
    if(q_t>obj[i].burst_time){
        sum=sum+obj[i].burst_time;
        g_c[i+1]=sum;
    }
    else if(q_t<obj[i].burst_time){
        a=obj[i].burst_time-q_t;
        sum=q_t+sum;
        g_c[i+1]=sum;
    }
    else if(obj[i].burst_time==0){
        a=a-q_t;
        sum=sum+q_t;
        g_c[i+1]=sum;
    }

    cout<<g_c[i]<<"  ";
}

w_t[0]=0;
t_t[0]=g_c[1];
for(int i=1;i<n;i++){
    t_t[i]=t_t[i-1]+obj[i].burst_time;
```

```cpp
            w_t[i]=t_t[i-1];
        }
    cout<<endl<<endl;
    for(int i=0;i<n;i++){
        tot_t=tot_t+t_t[i];
        wait_t=wait_t+w_t[i];
        cout<<w_t[i]<<"   "<<t_t[i]<<endl;
    }
    cout<<endl<<endl;
    cout<<"Avg. waiting time: "<<float(wait_t)/n<<endl;
    cout<<"Avg. turnaround time: "<<float(tot_t)/n<<endl;
}
```

Screenshot: