Student Name: Rahul Rathore
Student ID : 11702112
Email Address : rathodrahul873@gmail.com
GitHub Link: https://github.com/raahulrathore/osca3

Description:

The problem I was assigned was about scheduling of processes.The scheduling of processes is done to assign the CPU resource to the upcoming processes.Scheduling is done in operating systems to make out the better use of the CPU and other resources, to improve the efficiency and also reduce the delays and the wait time among the processes.In Technical terms , scheduling is done to fairly assign the CPU among the processes and deal with them without causing them delays.Various processes would be coming at different time intervals and CPU would be allocated to them fairly and all of them would be executed in finite amount of time.The scheduling algorithm I used is of Shortest Remaining Time First where the process with least amount of Execution time left will be allocated the CPU resource and would be completed first followed by next processes in the same manner.

Algorithm:

Shortest Remaining Time First :

As the name suggest, It works in the same manner.

1. Get all the Burst time.
2. Compare the arrival time of all the processes
3. Execute the process with the minimum remaining time first.

Description (Purpose of Use):

The purpose of the problem is to create a scheduling program which is capable of scheduling processes coming up at some time interval and use the CPU.

The constraints given were :

1. The CPU is not allocated to them more than 10 time units to any of the process
2. CPU is idle in the initial 3 time units
3. Scheduler must maintain a queue of processes.
4. CPU must execute process with shortest execution time first.
5. Arrival time must be known in prior to to allocate the resource.

Code Snippet:

Student Name: Rahul Rathore
Student ID : 11702112
Email Address : rathodrahul873@gmail.com
GitHub Link: https://github.com/raahulrathore/osca3

Code Snippet to indicate the part of code maintaining constraints :

```
int time=4,y=1; //cpu idle for first 3 time units
while(pl!=0)
{
    if(px!=0)
    {
    for(int i=0;i<n;i++)
    {
        if(p[i].arr>3 && p[i].arr<=time && p[i].status==0)          //inside of the queue
        {
            if(start==NULL)
            {
                p[i].status=1;
                cn=new node;
                cn->q=&p[i];
                cn->next=NULL;
                last=cn;
                start=cn;
            }
            else{
                p[i].status=1;
                cn=new node;
                cn->q=&p[i];
                node *x;
                x=start;
                if((start->q->burl)>(cn->q->burl))
                {
                    cn->next=start;
                    start=cn;
                }
                else
                {
```

Description of Boundary condition:

The code was having several boundry conditions to be kept in mind befire execution :

1. The burst time can't be greater than 10 time units for any of the process at any given point of time.
2. The CPU remains idle for first 3 time unit and only entertains processes prior to that.
3. Arrival time must not be lesser than 3 seconds as the CPU won't entertain that process.
4. Process with minimum value of burst time is to be executed first.
5. Average of the sum of all the waiting time is to be computed.
6. Average turn around time for all the processes is also computed.
7. Arrival time and Burst time of all the processes is to be known in prior , I.e before the allocation of resources.

Student Name: Rahul Rathore
Student ID : 11702112
Email Address : rathodrahul873@gmail.com
GitHub Link: https://github.com/raahulrathore/osca3

Description of Test Cases :

Following Test Cases were checked on the program and it passed all of them without any kind of ambiguity and exception :

1. Burst Time < 10 time units
2. Scheduler maintains a queue
3. CPU idle for first 3 time units
4. Process with minimum time remaining for execution implemented first

Have you made minimum 5 revisions of solution on GitHub?

Yes, whenever I was getting time from my schedule in the week , I used to work upon my code to make it more enhanced and smooth in respect of the problem assigned.

CODE :

Solution of the Problem assigned is as below :

```
#include <iostream>
#include<stdlib.h>

struct process{
	int arr,burst,firstt,i,status,burl,compt,waitingt,turnaroundt;
}*run=NULL;

int n,tq=10,px=0;
struct node
{
	struct process *q;
	struct node *next;
}*start=NULL,*last,*cn;

int main()
{
	printf("\nThe Number of the processes to be executed :");
	fflush(stdout);
	scanf("%d",&n);
	fflush(stdin);
	int pl=n;    //processes to be created
```

Student Name: Rahul Rathore
Student ID : 11702112
Email Address : rathodrahul873@gmail.com
GitHub Link: https://github.com/raahulrathore/osca3

```
struct process p[n];
for(int i=0;i<n;i++)
{
    printf("\nGive Arrival and Burst Time of processes %d ",i+1);
    fflush(stdout);
    printf("\n\tArrival Time : ");
    fflush(stdout);
    scanf("%d",&p[i].arr);
    fflush(stdin);
    printf("\tBurst Time : ");
    scanf("%d",&p[i].burst);
    fflush(stdin);
    p[i].burl=p[i].burst;
    p[i].i=i+1;
    p[i].status=0;
    p[i].firstt=-1;
    if(p[i].arr>3)
        px++;
}

int time=4,y=1; //cpu idle for first 3 time units
while(pl!=0)
{
    if(px!=0)
    {
    for(int i=0;i<n;i++)
    {
        if(p[i].arr>3 && p[i].arr<=time && p[i].status==0)
//inside of the queue
        {
            if(start==NULL)
            {
                p[i].status=1;
                cn=new node;
                cn->q=&p[i];
                cn->next=NULL;
                last=cn;
                start=cn;
            }
            else{
```

Student Name: Rahul Rathore
Student ID : 11702112
Email Address : rathodrahul873@gmail.com
GitHub Link: https://github.com/raahulrathore/osca3

```
                        p[i].status=1;
                        cn=new node;
                        cn->q=&p[i];
                        node *x;
                        x=start;
                        if((start->q->burl)>(cn->q->burl))
                        {
                              cn->next=start;
                              start=cn;
                        }
                        else
                        {
                              while(x->next!=NULL && x->next->q->burl <
cn->q->burl)
                              {
                                    x=x->next;
                              }
                              cn->next=x->next;
                              x->next=cn;
                        }
                  }
                  px--;
            }
      }
      }
      if(px==0 && y==1)
      {
            for(int i=0;i<n;i++)
            {
                  if(p[i].arr<=3 && p[i].status==0)
//in the queue
                  {
                        if(start==NULL)
                        {
                              p[i].status=1;
                              cn=new node;
                              cn->q=&p[i];
                              cn->next=NULL;
                              last=cn;
                              start=cn;
                        }
```

Student Name: Rahul Rathore
Student ID : 11702112
Email Address : rathodrahul873@gmail.com
GitHub Link: https://github.com/raahulrathore/osca3

```
                        else{
                            p[i].status=1;
                            cn=new node;
                            cn->q=&p[i];
                            node *x;
                            x=start;
                            if((start->q->burl)>(cn->q->burl))
                            {
                                cn->next=start;
                                start=cn;
                            }
                            else
                            {
                                while(x->next!=NULL && x->next->q->burl <
cn->q->burl)
                                {
                                    x=x->next;
                                }
                                cn->next=x->next;
                                x->next=cn;
                            }
                        }
                    }
                }
                y--;
            }
            if(run==NULL)     //process execution
            {
                if(start!=NULL)
                {
                    run=start->q;
                    start=start->next;
                    if(run->firstt==-1)
                    {
                        run->firstt=time;
                    }
                    if(run->burl<=tq)
                    {
                        printf("\nProcess P%d running from time= %d sec to time=
%d sec.",run->i,time,time+run->burl);
                        fflush(stdout);
```

Student Name: Rahul Rathore
Student ID : 11702112
Email Address : rathodrahul873@gmail.com
GitHub Link: https://github.com/raahulrathore/osca3

```
                        time=time+run->burl;
                        run->burl=0;
                        run->compt=time;
                        run->status=2;
                        pl--;
                    }
                    else if(run->burl>tq)
                    {
                        printf("\nProcess P%d running from time= %d sec to time=
%d sec.",run->i,time,time+tq);
                        fflush(stdout);
                        time=time+tq;
                        run->burl=run->burl-tq;
                        run->compt=time;
                        if(start==NULL)
                        {
                            cn=new node;
                            cn->q=run;
                            cn->next=NULL;
                            last=cn;
                            start=cn;
                        }
                        else
                        {
                            cn=new node;
                            cn->q=run;
                            node *x;
                            x=start;
                            if((start->q->burl)>(cn->q->burl))
                            {
                                cn->next=start;
                                start=cn;
                            }
                            else
                            {
                                while(x->next!=NULL && x->next->q->burl <
cn->q->burl)
                                {
                                    x=x->next;
                                }
                                cn->next=x->next;
```

Student Name: Rahul Rathore
Student ID : 11702112
Email Address : rathodrahul873@gmail.com
GitHub Link: https://github.com/raahulrathore/osca3

```
                                        x->next=cn;
                                    }
                                }
                            }
                        }
                        run=NULL;
                    }
                }

                int awt=0,att=0;
                for(int i=0;i<n;i++)
                {
                    p[i].turnaroundt=((p[i].compt)-(p[i].arr));
                    p[i].waitingt=((p[i].turnaroundt)-(p[i].burst));
                    awt=awt+p[i].waitingt;
                    att=att+p[i].turnaroundt;
                }
                awt=awt/n;
                att=att/n;
                printf("\n\n\nResult after scheduling    :");
                fflush(stdout);
                printf("\n
_____
_____");
                printf("\n|\tPROCESS\t\t|\tARRIVAL TIME\t|\tBURST TIME\t|\tWAITING
TIME    \t|\tTURN AROUND TIME    \t|");

printf("\n|_____|_____|_____|_____
_____|_____|");
                fflush(stdout);
                for(int i=0;i<n;i++)
                {

printf("\n|\t\tP%d\t\t|\t\t%d\t\t\t|\t\t%d\t\t|\t\t\t%d\t\t|\t\t\t%d\t\t\t|",p[i].i,p[i].arr,p[i].bur
st,p[i].waitingt,p[i].turnaroundt);

printf("\n|_____|_____|_____|_____
_____|_____|");
                }

                printf("\nAverage    waiting time = %d ",awt);
```

Student Name: Rahul Rathore
Student ID : 11702112
Email Address : rathodrahul873@gmail.com
GitHub Link: https://github.com/raahulrathore/osca3

```
    fflush(stdout);
    printf("\nAverage Turn Around Time = %d \n",att);
}
```

GITHUB LINK FOR THE REPO : https://github.com/raahulrathore/osca3