

## Lab Report: 04

**Report Title:** SJF, Priority Scheduling, Shell Script Programs

**Name:** Md. Rakibur Rahman, **Batch:** D-59, **Roll:** 30, **Course Code:** CSE-310

### C++ Program of Shortest-Job-First (SJF) scheduling

```
#include<iostream>
using namespace std;
int main()
{
    int a[10],b[10],x[10];
    int waiting[10],turnaround[10],completion[10];
    int i,j,smallest,count=0,time,n;
    double avg=0,tt=0,end;

    cout<<"Enter the number of Processes: "; //input
    cin>>n;
    cout<<"\n";
    for(i=0; i<n; i++)
    {
        cout<<"Enter arrival time for p"<<i+1<<": "; //input
        cin>>a[i];
    }
    cout<<"\n";
    for(i=0; i<n; i++)
    {
        cout<<"Enter burst time for p"<<i+1<<": "; //input
        cin>>b[i];
    }
    for(i=0; i<n; i++)
        x[i]=b[i];

    b[9]=9999;
    for(time=0; count!=n; time++)
    {
        smallest=9;
        for(i=0; i<n; i++)
        {
            if(a[i]<=time && b[i]<b[smallest] && b[i]>0 )
                smallest=i;
        }
        b[smallest]--;
```

```

        if(b[smallest]==0)
        {
            count++;
            end=time+1;
            completion[smallest] = end;
            waiting[smallest] = end - a[smallest] - x[smallest];
            turnaround[smallest] = end - a[smallest];
        }
    }
    cout<<"\n";
    cout<<"Process"<<"\t"<< "burst-time"<<"\t"<<"arrival-time" <<"\t"<<"waiting-time"
<<"\t"<<"turnaround-time"<< "\t"<<"completion-time"<<endl;
    for(i=0; i<n; i++)
    {
        cout<<"p"<<i+1<<"\t\t"<<x[i]<<"\t\t"<<a[i]<<"\t\t"<<waiting[i]<<"\t\t"<<turnaround[i]<<"\t\t"<<compl
        etion[i]<<endl;
        avg = avg + waiting[i];
        tt = tt + turnaround[i];
    }
    cout<<"\n\nAverage waiting time = "<<avg/n <<"ms";
    cout<<" Average Turnaround time = "<<tt/n<< "ms" <<endl;
}

```

### Output

```

Enter the number of process: 5

Enter arrival time for p1: 3
Enter arrival time for p2: 1
Enter arrival time for p3: 4
Enter arrival time for p4: 0
Enter arrival time for p5: 2

Enter burst time for p1: 1
Enter burst time for p2: 4
Enter burst time for p3: 2
Enter burst time for p4: 6
Enter burst time for p5: 3

Process  Arrival-time  Burst-time  Waiting-time  Turnaround-time
p1        3             1             3             4
p2        4             2             3             5
p3        2             3             7            10
p4        1             4            11            15
p5        0             6             0             6

Average Waiting Time = 4.8ms
Average Turnaround Time = 8ms

```

### C++ Program of priority-scheduling

```
#include<iostream>
using namespace std;
int main(){
    int a[10],b[10],x[10];
    int waiting[10],turnaround[10],completion[10],p[10];
    int i,j,smallest,count=0,time,n;
    double avg=0,tt=0,end;
    cout<<"Enter the number of Processes: ";
    cin>>n;
    cout<<"\n";
    for(i=0;i<n;i++) {
        cout<<"Enter arrival time for p"<<i+1<<": ";
        cin>>a[i];
    }
    cout<<"\n";
    for(i=0;i<n;i++) {
        cout<<"Enter burst time for p"<<i+1<<": ";
        cin>>b[i];
    }
    cout<<"\n";
    for(i=0;i<n;i++) {
        cout<<"Enter priority for p"<<i+1<<": ";
        cin>>p[i];
    }
    for(i=0; i<n; i++)
        x[i]=b[i];

    p[9]=-1;
    for(time=0; count!=n; time++){
        smallest=9;
        for(i=0; i<n; i++) {
            if(a[i]<=time && p[i]>p[smallest] && b[i]>0 )
                smallest=i;
        }
        b[smallest]--;

        if(b[smallest]==0) {
            count++;
            end=time+1;
            completion[smallest] = end;
            waiting[smallest] = end - a[smallest] - x[smallest];
            turnaround[smallest] = end - a[smallest];
        }
    }
}
```

```

    }
}
cout<<"\n";
cout<<"Process"<<"\t"<<"burst-time"<<"\t"<<"arrival-time"<<"\t"<<"waiting-time"
<<"\t"<<"turnaround-time"<<"\t"<<"completion-time"<<"\t"<<"Priority"<<endl;
for(i=0; i<n; i++)
{
    cout<<"p"<<i+1<<"\t\t"<<x[i]<<"\t\t"<<a[i]<<"\t\t"<<waiting[i]<<"\t\t"<<turnaround[i]<<"\t\t"<<compl
    etion[i]<<"\t\t"<<p[i]<<endl;
    avg = avg + waiting[i];
    tt = tt + turnaround[i];
}
cout<<"\n\nAverage waiting time ="<<avg/n<<"ms\n";
cout<<"Average Turnaround time ="<<tt/n<<"ms"<<endl;
}

```

## Output

```

C:\Users\Lenovo\OneDrive\Desktop\sjf.exe
Enter the number of Processes: 5

Enter arrival time for p1: 0
Enter arrival time for p2: 1
Enter arrival time for p3: 2
Enter arrival time for p4: 3
Enter arrival time for p5: 4

Enter burst time for p1: 4
Enter burst time for p2: 3
Enter burst time for p3: 1
Enter burst time for p4: 5
Enter burst time for p5: 2

Enter priority for p1: 2
Enter priority for p2: 3
Enter priority for p3: 4
Enter priority for p4: 5
Enter priority for p5: 5

Process burst-time    arrival-time    waiting-time    turnaround-time    completion-time    Priority
p1          4          0          11          15          15          2
p2          3          1          8          11          12          3
p3          1          2          0          1          3          4
p4          5          3          0          5          8          5
p5          2          4          4          6          10         5

Average waiting time =4.6ms
Average Turnaround time =7.6ms

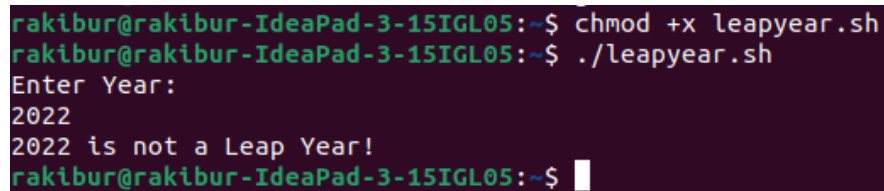
```

## Linux Shell Scripts

### 1. Shell Script to Check Leap Year

```
echo "Enter Year:"
read y
year=$y
y=$(( $y % 4 ))
if [ $y -eq 0 ]
then
    echo "$year is Leap Year!"
else
    echo "$year is not a Leap Year!"
fi
```

#### Output



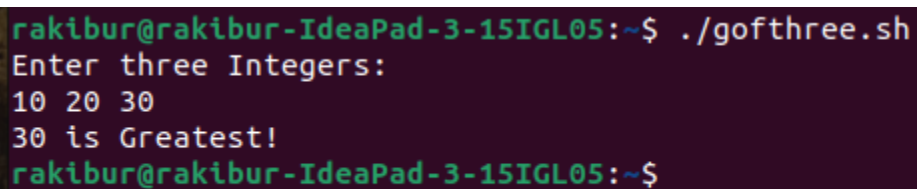
```
rakibur@rakibur-IdeaPad-3-15IGL05:~$ chmod +x leapyear.sh
rakibur@rakibur-IdeaPad-3-15IGL05:~$ ./leapyear.sh
Enter Year:
2022
2022 is not a Leap Year!
rakibur@rakibur-IdeaPad-3-15IGL05:~$
```

### 2. Shell Script to find the greatest of three numbers

```
echo "Enter three Integers:"
read a b c

if [ $a -gt $b -a $a -gt $c ]
then
    echo "$a is Greatest"
elif [ $b -gt $c -a $b -gt $a ]
then
    echo "$b is Greatest"
else
    echo "$c is Greatest!"
fi
```

#### Output

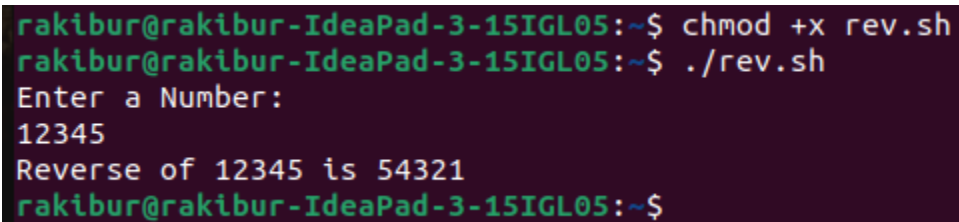


```
rakibur@rakibur-IdeaPad-3-15IGL05:~$ ./gofthree.sh
Enter three Integers:
10 20 30
30 is Greatest!
rakibur@rakibur-IdeaPad-3-15IGL05:~$
```

### 3. Shell Script to Reverse a Number

```
echo "Enter a Number:"
read a
rev=0
sd=0
or=$a
while [ $a -gt 0 ]
do
    sd=`expr $a % 10`
    temp=`expr $rev \* 10`
    rev=`expr $temp + $sd`
    a=`expr $a / 10`
done
echo "Reverse of $or is $rev"
```

#### Output



```
rakibur@rakibur-IdeaPad-3-15IGL05:~$ chmod +x rev.sh
rakibur@rakibur-IdeaPad-3-15IGL05:~$ ./rev.sh
Enter a Number:
12345
Reverse of 12345 is 54321
rakibur@rakibur-IdeaPad-3-15IGL05:~$
```

### 4. Shell Script to Check Number is Positive or Negative

```
echo "Enter a Number:"
read n
if [ $n -gt 0 ]
then
    echo "Number is positive"
else
    if [ $n -eq 0 ]
    then
        echo "Number is Zero!"
    else
        echo "Number is Negative!!"
    fi
fi
```

## Output

```
rakibur@rakibur-IdeaPad-3-15IGL05:~$ chmod +x posnev.sh
rakibur@rakibur-IdeaPad-3-15IGL05:~$ ./posnev.sh
Enter a Number:
5
Number is positive
rakibur@rakibur-IdeaPad-3-15IGL05:~$ ./posnev.sh
Enter a Number:
-3
Number is Negative!!
rakibur@rakibur-IdeaPad-3-15IGL05:~$ ./posnev.sh
Enter a Number:
0
Number is Zero!
rakibur@rakibur-IdeaPad-3-15IGL05:~$
```

## 5. Shell Script to Print Fibonacci Series

```
c=0
a=1
b=1

read -p "Enter limit of Fibonacci Series:" n
echo -n "$a "
echo -n "$b "
#Fibonacci series logic
while((c<n))
do
c=$((a+b))
echo -n "$c "          # To sort a file numerically used -n option.
a=$b
b=$c
done
#echo -e "\n"
```

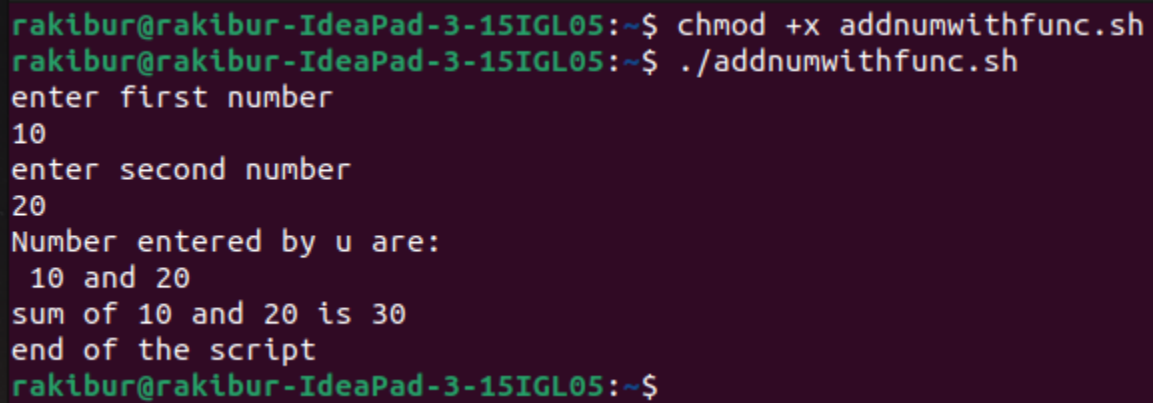
## Output

```
rakibur@rakibur-IdeaPad-3-15IGL05:~$ chmod +x fibonacci.sh
rakibur@rakibur-IdeaPad-3-15IGL05:~$ ./fibonacci.sh
Enter limit of Fibonacci Series:6
1 1 2 3 5 8 rakibur@rakibur-IdeaPad-3-15IGL05:~$
```

## 6. Shell Script to add two numbers by the use of Function

```
#function to add two numbers
add()
{
x=$1
y=$2
echo -e "Number entered by u are:\n $x and $y"
echo "sum of $1 and $2 is `expr $x + $y` "
}
# main script
echo "enter first number"
read first
echo "enter second number"
read sec
#calling function
add $first $sec
echo "end of the script"
```

### Output



```
rakibur@rakibur-IdeaPad-3-15IGL05:~$ chmod +x addnumwithfunc.sh
rakibur@rakibur-IdeaPad-3-15IGL05:~$ ./addnumwithfunc.sh
enter first number
10
enter second number
20
Number entered by u are:
 10 and 20
sum of 10 and 20 is 30
end of the script
rakibur@rakibur-IdeaPad-3-15IGL05:~$
```