LOVELY PROFESSIONAL UNIVERSITY

Academic Task-3

(Operating System)

School of Computer Science and Engineering(Faculty of Technology And Sciences)

Name of the faculty member: Ashu

Course Code: **CSE 316** **Course Title:** Operating System

Term:219202

Date of Allotment: 01/03/2020 Date of Submission: 06/03/2020

Name: Kousik Snai
Reg No: 11804893
Roll No: 45
Section: K18UW
Email Address: kousiksnaiscience@gmail.com
GitHub Link: https://github.com/kousiksnai99/Operating_Sysytem_Project

Problem 1:

Consider that a system has P resources of same type. These resources are shared by Q processes time to time.

All processes request and release the resources one at a time. Generate a solution to demonstrate that,

the system is in safe state when following conditions are satisfied.

Conditions:

1. Maximum resource need of each process is between 1 and P.
2. Summation of all maximum needs is less than P+Q

Description::

This problem is based on Round Robin Algorithm. In this problem we are taking input from user and user can enter limited no of jobs.

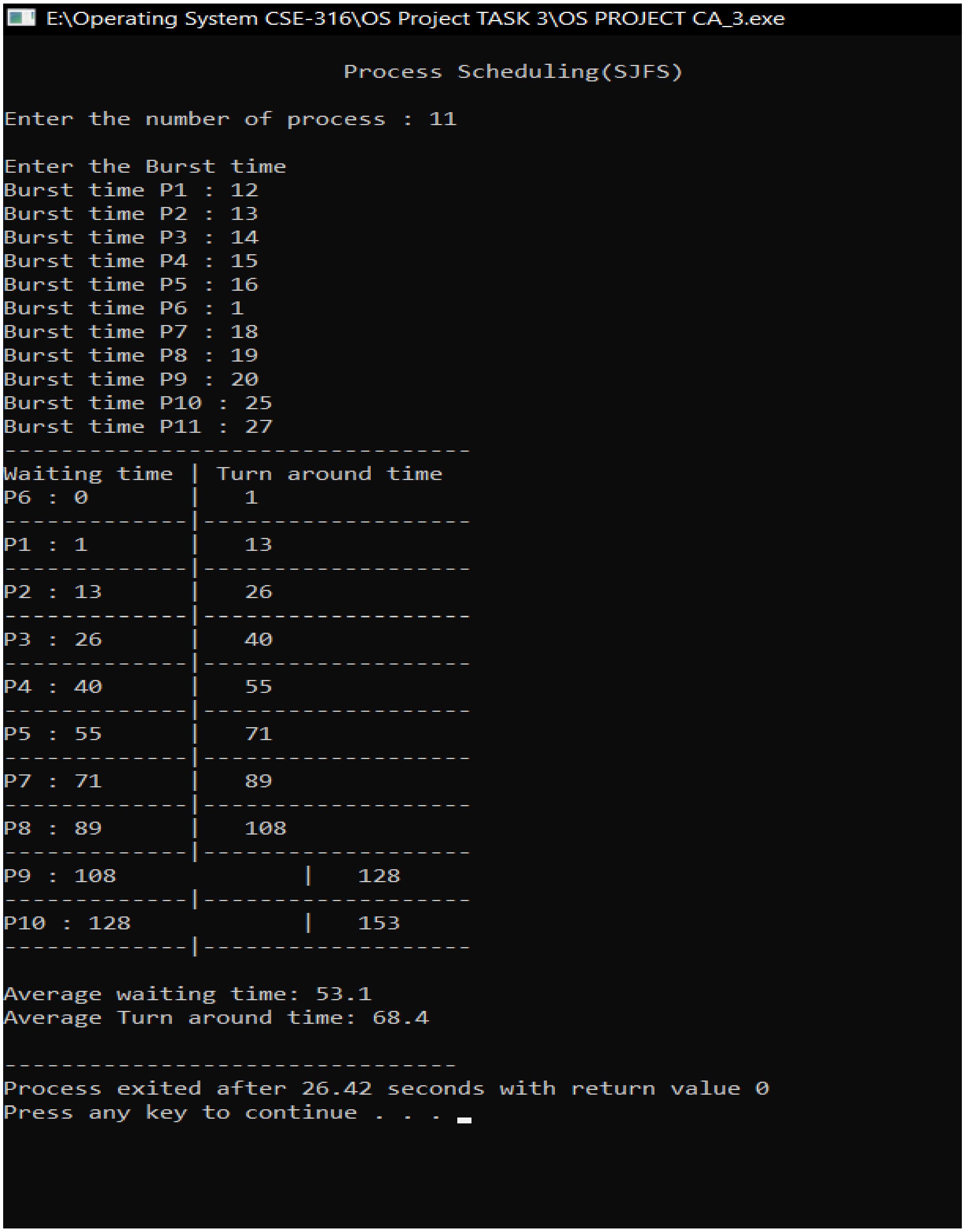
While solving this first we take value from user and store it in a queue. The value of front and rear used here is -1 as initial they are not at any position. Now process run in queue and if its burst time is less than time given quantum then this process will be completed and we take out that process from queue. If time quantum is less than burst time then the process will again enter in the queue. This will happen till all process are completed.

Alogithm::

```
#include<bits/stdc++.h>

using namespace std;

int main()

{

 int n,i,j,avgw=0;

 float avgt=0,temp;

 int burst[10],wait[10],process[10];

 cout<<"\n\t\t\tProcess Scheduling(SJFS)\n";

 cout<<"\nEnter the number of process : ";

 cin>>n;

 cout<<"\nEnter the Burst time\n";


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

 {

            cout<<"Burst time P"<<i+1<<" : ";

    cin>>burst[i];

    process[i] = i;

 }


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


 {

   for(j=i;j<n;j++)


    {

      if(burst[i]>burst[j])


        {

        temp = burst[i];
        burst[i] = burst[j];

        burst[j] = temp;

        temp = process[i];

        process[i] = process[j];

        process[j] = temp;
      }
   }
 }
 cout<<"---------------------------------\n";
 cout<<"Waiting time | Turn around time \n";

j=0;

for(i=0;i<n;i++)
 {
    wait[i] = j;
    j = burst[i] + j;
    cout<<"P"<<process[i]+1<<" : "<<wait[i]<<"\t     |   "<<wait[i]+burst[i]<<"\n";
    cout<<"-------------|-------------------\n";
    avgw+=wait[i];
    avgt+=wait[i]+burst[i];
 }
 cout<<"\n";
 cout<<"Average waiting time: "<<avgw/(float)n<<"\n";
 cout<<"Average Turn around time: "<<avgt/(float)n<<"\n";
 return 0;
```

}

output:



```
E:\Operating System CSE-316\OS Project TASK 3\OS PROJECT CA_3.exe

                                Process Scheduling(SJFS)

Enter the number of process : 11

Enter the Burst time
Burst time P1 : 12
Burst time P2 : 13
Burst time P3 : 14
Burst time P4 : 15
Burst time P5 : 16
Burst time P6 : 1
Burst time P7 : 18
Burst time P8 : 19
Burst time P9 : 20
Burst time P10 : 25
Burst time P11 : 27
---------------------------------
Waiting time | Turn around time
P6 : 0       |   1
-------------|-------------------
P1 : 1       |   13
-------------|-------------------
P2 : 13      |   26
-------------|-------------------
P3 : 26      |   40
-------------|-------------------
P4 : 40      |   55
-------------|-------------------
P5 : 55      |   71
-------------|-------------------
P7 : 71      |   89
-------------|-------------------
P8 : 89      |   108
-------------|-------------------
P9 : 108             |   128
-------------|-------------------
P10 : 128            |   153
-------------|-------------------

Average waiting time: 53.1
Average Turn around time: 68.4

--------------------------------
Process exited after 26.42 seconds with return value 0
Press any key to continue . . .
```