

PRACTICAL 9

AIM: Implement Two server(multi) queuing system.

THEORY:

The system consists of multiple servers and a common queue for all items. When any item requests for the server, it is allocated if at-least one server is available. Else the queue begins to start until the server is free. In this system, we assume that all servers are identical, i.e. there is no difference which server is chosen for which item. There is an exception of utilization.

Multi server queue has two or more service facilities in parallel providing identical service. All the customers in the waiting line can be served by more than one station. The arrival time and the service time follow poisson and exponential distribution.

PROGRAM:

```
#include<bits/stdc++.h>
using namespace std;
constexpr int FLOAT_MIN = 0;
constexpr int FLOAT_MAX = 1;
int main()
{
    std::random_device rd;
    std::default_random_engine eng(rd());
    std::uniform_real_distribution<float> distr(FLOAT_MIN, FLOAT_MAX);
    float r, iat, clock=0, nat, it1, it2, run=150, cit1=0, cit2=0;
    float mean, lemدا1, lemدا2;
    cout<<"enter mean time: ";
    cin>>mean;
    cout<<"service time of server1: ";
    cin>>lemدا1;
    cout<<"service time of server2: ";
    cin>>lemدا2;
    float se1=0, se2=0;
    int k, q=0, qmax=3, kont=0, counter;
    printf("\n CLOCK    IAT    NAT    SE1    SE2    QUE    KONT    CIT1    CIT2");
    r=distr(eng);
    iat=(-mean)*log(1-r);
    nat=nat+iat;
    se1=lemدا1;
```

```

counter=1;

printf("\n %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %d  %d  %6.2f  %6.2f
",clock,iat,nat,se1,se2,q,kont,cit1,cit2);
while(clock<=run)
{
    if(nat<=se1 && nat<=se2)
    {
        clock=nat;
        q=q+1;
        r=distr(eng);
        iat = (-mean)*log(1-r);
        nat=nat+iat;
        counter=counter+1;
    }
    else if (se1<=nat && se1<=se2)
        clock=se1;
    else
        clock=se2;

    if(q>qmax)
    {
        kont=kont+1;
        q=q-1;
    }
    else if(q>=1 && se1<=clock)
    {
        it1=clock-se1;
        cit1=cit1+it1;
        se1=clock+lemda1;
        q=q-1;
    }

    else if(q>=1 && se2<=clock)
    {
        it2=clock-se2;
        cit2=cit2+it2;
        se2=clock+lemda2;
        q=q-1;
    }
}

```

```

else if(q==0&& se1<=clock)
{
    clock=nat;
    it1=clock-se1;
    cit1=cit1+it1;
    se1=nat+lemda1;
    r=distr(eng);
    iat=(-mean)*log(1-r);
    nat=nat+iat;
    counter=counter+1;
}
else if(q==0 && se2<=clock)
{
    clock=nat;
    it2=clock-se2;
    cit2=cit2+it2;
    se2=nat+lemda2;
    r=distr(eng);
    iat=(-mean)*log(1-r);
    nat=nat+iat;
    counter=counter+1;
}
    printf("\n %6.2f  %6.2f  %6.2f  %6.2f  %6.2f  %d  %d  %6.2f  %6.2f",clock,iat,nat,se1,se2,q,kont,cit1,cit2);
}

printf("\n clock=%8.2f  cit1=%6.2f  cit2=%6.2f  counter=%d",clock,cit1,cit2,counter);
printf("\n\n Mean arrival time = %5.2f minutes exponentially distributed",mean);
printf("\n Service time : \nServer1=%5.2f minutes\nServer2=%5.2f minutes",lemda1,lemda2);
printf("\nSimulation run(Elapsed time)=%7.2f minutes",clock);
printf("\nNumber of customers arrived=%d",counter);
printf("\nNumber of customers returned without service=%d",kont);
printf("\nIdle time of server1 = %6.2f minutes",cit1);
printf("\nIdle time of server2 = %6.2f minutes\n",cit2);
}

```

OUTPUT:

```

"C:\Users\Ankit Goyal\OneDrive\Documents\labs\8th Sem Lab\SSM\twoServer.exe"

enter mean time: 9
service time of server1: 8
service time of server2: 10

CLOCK      IAT      NAT      SE1      SE2      QUE      KONT      CIT1      CIT2
0.00        1.17      1.17      8.00      0.00      0         0         0.00      0.00
1.17        10.46     11.64     8.00      11.17     0         0         0.00      1.17
11.64       6.49      18.13     19.64     11.17     0         0         3.64      1.17
18.13       5.75      23.87     19.64     28.13     0         0         3.64      8.13
23.87       9.34      33.22     31.87     28.13     0         0         7.87      8.13
33.22       3.92      37.14     31.87     43.22     0         0         7.87      13.22
37.14       3.74      40.88     45.14     43.22     0         0         13.14     13.22
40.88       2.50      43.38     45.14     43.22     1         0         13.14     13.22
43.22       2.50      43.38     45.14     53.22     0         0         13.14     13.22
43.38       3.12      46.50     45.14     53.22     1         0         13.14     13.22
45.14       3.12      46.50     53.14     53.22     0         0         13.14     13.22
46.50       1.25      47.76     53.14     53.22     1         0         13.14     13.22
47.76       13.57     61.33     53.14     53.22     2         0         13.14     13.22
53.14       13.57     61.33     61.14     53.22     1         0         13.14     13.22
53.22       13.57     61.33     61.14     63.22     0         0         13.14     13.22
61.33       8.32      69.65     69.33     63.22     0         0         13.33     13.22
69.65       3.03      72.68     69.33     79.65     0         0         13.33     19.65
72.68       8.45      81.13     80.68     79.65     0         0         16.68     19.65
81.13       1.40      82.53     80.68     91.13     0         0         16.68     21.13
82.53       4.55      87.09     90.53     91.13     0         0         18.53     21.13
87.09       1.46      88.55     90.53     91.13     1         0         18.53     21.13
88.55       7.84      96.38     90.53     91.13     2         0         18.53     21.13
90.53       7.84      96.38     98.53     91.13     1         0         18.53     21.13
91.13       7.84      96.38     98.53     101.13    0         0         18.53     21.13
96.38       3.80      100.18    98.53     101.13    1         0         18.53     21.13
98.53       3.80      100.18    106.53    101.13    0         0         18.53     21.13
100.18      10.75     110.92    106.53    101.13    1         0         18.53     21.13
101.13      10.75     110.92    106.53    111.13    0         0         18.53     21.13
110.92      9.98      120.90    118.92    111.13    0         0         22.92     21.13
120.90      20.42     141.32    118.92    130.90    0         0         22.92     30.90
141.32      19.19     160.51    149.32    130.90    0         0         45.32     30.90
160.51      7.81      168.32    149.32    170.51    0         0         45.32     60.51
clock= 160.51 cit1= 45.32 cit2= 60.51 counter=24

Mean arrival time = 9.00 minutes exponentially distributed
Service time :
Server1= 8.00 minutes
Server2=10.00 minutes
Simulation run(Elapsed time)= 160.51 minutes
Number of customers arrived=24
Number of customers returned without service=0
Idle time of server1 = 45.32 minutes
Idle time of server2 = 60.51 minutes

Process returned 0 (0x0)   execution time : 14.839 s
Press any key to continue.

```