

УДК 519.85.004.41

*Постова С. А. канд, пед. наук, магістрант, гр. ЗПІ-18-2м,
Науковий керівник: Колос К.Р., д-р. пед. наук, професор кафедри
Житомирський державний технологічний університет*

МОДЕЛЮВАННЯ РОБОТИ ДЕТЕРМІНОВАНИХ ОДНОКАНАЛЬНИХ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ ЗА ДОПОМОГОЮ МОВИ C++

Одноканальна система масового обслуговування (СМО) характеризується тим, що для обслуговування заявок наявний тільки один пристрій. Для кожної СМО обов'язково потрібно вказати порядок вибору заявок з черги на обслуговування, тобто дисципліну обслуговування (ДО). ДО FIFO характеризується тим, що заявки на обслуговування обираються з черги в порядку надходження.

Розглянемо покроковий алгоритм моделювання роботи одноканальної СМО з дисципліною обслуговування FIFO для безпріоритетних заявок та 1 позицією черги:

1. Заявка, надходить до 1-канальної СМО, перевіряє чи вільний пристрій.

2. Якщо вільний, то одразу обслуговується. Час виходу такої заявки дорівнює: час входу + час обслуговування.

3. Якщо пристрій зайнятий, то заявка перевіряє чи може стати в чергу. Якщо так, то стоїть в черзі до тих пір, поки не звільниться пристрій. Коли пристрій звільняється, то заявка переходить до нього й обслуговується. Час виходу такої заявки дорівнює: час входу + час очікування + час обслуговування.

4. Якщо місця в черзі немає, то заявка залишає СМО не обслуженою. Час виходу такої заявки дорівнює часу входу.

Нехай задано час входу та час, необхідний для обслуговування 5 заявок (табл 1.)

Таблиця 1. – Початкові дані для розв'язанні задачі

№ заявки	Час входу	Час, необхідний для обслуговування
1.	3	2
2.	5	4
3.	1	4
4.	7	6
5.	9	1

Використовуючи принципи дискретно подійного моделювання знайдемо час виходу для кожної заявки. Для цього потрібно побудувати

діаграму простору станів, що відображає реальний стан функціонування СМО.

Перед початком розв'язання задачі та побудови простору станів потрібно впорядкувати заявки за їхнім часом надходження (табл. 2),

Таблиця 2. – Впорядковані початкові дані

№ заявки	Час входу	Час, необхідний для обслуговування
1.	1	4
2.	3	2
3.	5	4
4.	7	6
5.	9	1

Перейдемо до побудови простору станів. Перша заявка знаходить до СМО на 1 секунді і перевіряє чи вільний пристрій, оскільки він в цей час вільний, пристрій приступає до її виконання і час виходу цієї заявки = час входу+час обслуговування = 5. На 3 секунді до СМО надходить 2 заявка та перевіряє чи вільний пристрій. В цей час пристрій зайнятий обслуговуванням першої заявки, тому заявка перевіряє чи може вона стати в чергу. В даний час черга вільна, тому заявка стоїть в черзі до звільнення пристрою. Одразу після того, як пристрій закінчив обслуговування першої заявки на 5 секунді, він приступив до обслуговування 2 заявки. Тому Час виходу 2 заявки = час входу + час очікування в черзі + час обслуговування = 7. На 5-ій секунді до СМО надходить третя заявка. В цей час пристрій зайнятий обслуговуванням другої заявки, а черга вільна, тому дана заявка стоїть в черзі 2 секунди, а потім надходить в пристрій для обслуговування. Час виходу третьої заявки = час входу + час очікування в черзі + час обслуговування = 11. На 7 секунді до системи надходить четверта заявка. В цей час пристрій зайнятий обслуговуванням третьої заявки, а черга вільна, тому дана заявка стоїть в черзі 4 секунди, а потім надходить в пристрій для обслуговування. Час виходу третьої заявки = час входу + час очікування в черзі + час обслуговування = 17. На 9 секунді до системи надходить п'ята заявка. В цей час пристрій обслуговує третю заявку і в черзі стоїть четверта заявка, тому п'ята заявка залишає СМО необслуженою і час її виходу=часу входу=9.

Розглянемо код програми мовою C++, що імітує роботу одноканальної системи масового обслуговування з 1 позицією чергою, притримуючись ДО FIFO, якщо час входу та час обслуговування кожної заявки вводяться з клавіатури:

```
#include <bits/stdc++.h>
using namespace std;
int main ()
```

```

{ int vhod [100], vyhod[100], prystr[1000], cherga[1000],obsl[100];
int i,j,k,l,n; bool flag;
cout<<"Input kol-vo zayavok"; cin>>n;
for (i=0;i<n; i++)
{ cout<<"Input t vhoda "<<i+1<<"-oi zayavki: "; cin>>vhod[i];
cout<<"Input t obsl "<<i+1<<"-oi zayavki: "; cin>>obsl[i]; vyhod[i]=0;}
for(int i = 1; i < n; ++i)
    for(int r = 0; r < n-i; r++)
        if(vhod[r] > vhod[r+1])
            {swap(vhod[r],vhod[r+1]); swap(obsl[r],obsl[r+1]); }
for (j=1;j<=1000;j++) { prystr[j]=0; cherga[j]=0; }
for (l=0;l<n; l++)
    { if (prystr[vhod[l]+1]>0 && cherga[vhod[l]+1]>0) vyhod[l]=vhod[l];
    if (prystr[vhod[l]+1]>0 && cherga[vhod[l]+1]==0)
        {k=vhod[l]+1;
        while (prystr[k]>0) {cherga[k]=l+1; k=k+1;}
        vhod[l]=k-1; }
    if (prystr[vhod[l]+1]==0)
        { for (j=vhod[l]+1; j<=vhod[l]+obsl[l];j++) prystr[j]=l+1;
        vyhod[l]=vhod[l]+obsl[l]; }
cout<<"T vyhoda "<<l+1<<"-oi zayavki = "<<vyhod[l]<<endl; }
return 0;}

```

```

Input kol-vo zayavok5
Input t vhoda 1-oi zayavki: 3
Input t obsl 1-oi zayavki: 2
Input t vhoda 2-oi zayavki: 5
Input t obsl 2-oi zayavki: 4
Input t vhoda 3-oi zayavki: 1
Input t obsl 3-oi zayavki: 4
Input t vhoda 4-oi zayavki: 7
Input t obsl 4-oi zayavki: 6
Input t vhoda 5-oi zayavki: 9
Input t obsl 5-oi zayavki: 1
T vyhoda 1-oi zayavki = 5
T vyhoda 2-oi zayavki = 7
T vyhoda 3-oi zayavki = 11
T vyhoda 4-oi zayavki = 17
T vyhoda 5-oi zayavki = 9

-----
Process exited after 12.58 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

Рис. 1. Результат роботи програми

Порівнявши дані, отримані в результаті роботи програми з даними, отриманими при побудові простору станів на основі дискретно-подвійного моделювання, можна зробити висновок, що запропонований алгоритм, реалізований мовою C++ вірно імітує роботу одноканальної СМО з дисципліною обслуговування FIFO та 1 позицією черги.