

Міністерство освіти і науки України
Національний університет «Запорізька Політехніка»

Кафедра програмних засобів

ЗВІТ

з лабораторної роботи №3

з дисципліни «Моделювання та Аналіз Програмного Забезпечення» на

тему:

«Моделювання системи масового обслуговування зі зворотнім зв'язком.

Моделювання багатоканальної системи масового обслуговування»

Виконав:

Студент групи КНТ-122

О. А. Онищенко

Прийняли:

Викладач:

Ж. К. Камінська

2024

МОДЕЛЮВАННЯ СИСТЕМИ МАСОВОГО ОБСЛУГОВУВАННЯ ЗІ ЗВОРОТНІМ ЗВ'ЯЗКОМ. МОДЕЛЮВАННЯ БАГАТОКАНАЛЬНОЇ СИСТЕМИ МАСОВОГО ОБСЛУГОВУВАННЯ

Мета роботи

Метою роботи є вивчення методів моделювання СМО зі зворотнім зв'язком та багатоканальних СМО на основі використання SIMC

Результати виконання

Код програми

```
#include "../simc/simc.h"
#include <iostream>
using namespace std;

void one() {
    auto Modeling_Hours=40;
    auto Total_Modeling_Time=Modeling_Hours*60;

    auto Assembly_Delay=30;
    auto Firing_Delay=8;

    auto Worker_Hourly_Salary=50;
    auto Firing_Daily_Price=200;
    auto Material_Price=2;
    auto Product_Price=7;

    auto Workers_Min=4;
    auto Workers_Max=6;

    auto Best_Result=0;
    auto Best_Count=0;

    pqueue Assembly_Queue;
    pstorage Assembly_Facility;
    pqueue Firing_Queue;
    pfacility Firing_Facility;

    initlist(Total_Modeling_Time);
    initcreate(1, 0);
```

```

newqueue(Assembly_Queue, "\"Assembly Queue\"");
newqueue(Firing_Queue, "\"Firing Queue\"");
newfac(Firing_Facility, "\"Firing Facility\"");

for (auto
Workers_Count=Workers_Min;Workers_Count<=Workers_Max;Workers_Count++) {
    for (auto j=Workers_Min;j<=Workers_Count;j++) initcreate(1,0);
    newstorage(Assembly_Facility, "\"Assembly Facility\"", 3);
    auto Parts_Assembled=0;
    while (systime<Total_Modeling_Time) {
        plan();
        switch (sysevent) {
            case 1: inqueue(Assembly_Queue); break;
            case 2: enter(Assembly_Facility, 1); break;
            case 3: outqueue(Assembly_Queue); break;
            case 4: delayt(Assembly_Delay); break;
            case 5: leave(Assembly_Facility, 1); break;

            case 6: inqueue(Firing_Queue); break;
            case 7: seize(Firing_Facility); break;
            case 8: outqueue(Firing_Queue); break;
            case 9: delayt(Firing_Delay); break;
            case 10: outfac(Firing_Facility); Parts_Assembled+=1; break;
            case 11: next(1); break;
        }
    }
    auto
Workers_Salary=Worker_Hourly_Salary*Modeling_Hours*Workers_Count;
    auto Firing_Facility_Cost=Firing_Daily_Price/8*Modeling_Hours;
    auto Materials_Cost=Parts_Assembled*Material_Price;
    auto Parts_Cost=Parts_Assembled*Product_Price;
    auto
Total_Expenses=Workers_Salary+Firing_Facility_Cost+Materials_Cost;
    auto Profit=Parts_Cost-Total_Expenses;
    if (abs(Profit)>abs(Best_Result)) { Best_Result=Profit;
Best_Count=Workers_Count; }
    cout << "Workers: " << Workers_Count << " Profit: " << Profit <<
endl;
}

    cout << "\nBest Count: " << Best_Count << " Best Result: " <<
Best_Result << endl << endl;
    printall();
}

void two() {
    auto Interval=115;

```

```

auto First_Delay=335;
auto Second_Delay=110;

auto Modeling_Hours=1;
auto Total_Modeling_Time=Modeling_Hours*60*60;

pqueue First_Queue;
pqueue Second_Queue;
pfacility First_Facility;
pfacility Second_Facility;

initlist(Total_Modeling_Time);
initcreate(1, 0);

newqueue(First_Queue, "\"First Queue\"");
newqueue(Second_Queue, "\"Second Queue\"");
newfac(First_Facility, "\"First Facility\"");
newfac(Second_Facility, "\"Second Facility\"");

while (systime<Total_Modeling_Time) {
    plan();
    switch (sysevent) {
        case 1: create(Interval); break;
        case 2: cout << "First Queue Length: " << First_Queue->lq << endl;
if (Second_Queue->status == queue::empty) next(8); else
inqueue(First_Queue); break;
        case 3: seize(First_Facility); break;
        case 4: outqueue(First_Queue); break;
        case 5: delayt(First_Delay); break;
        case 6: outfac(First_Facility); break;
        case 7: destroy(); break;

        case 8: create(Interval); break;
        case 9: cout << "Second Queue Length: " << Second_Queue->lq <<
endl; Second_Queue->mq=1; inqueue(Second_Queue); break;
        case 10: seize(Second_Facility); break;
        case 11: outqueue(Second_Queue); break;
        case 12: delayt(Second_Delay); break;
        case 13: outfac(Second_Facility); break;
        case 14: destroy(); break;
    }
}

cout << "\nFirst Queue Max Length: " << First_Queue->mq << " Average
Length: " << First_Queue->lm << endl << endl;
printall();
}

```

```

void three() {
    auto First_Interval=1;
    auto First_Delay=1;

    auto Second_Interval=4;
    auto Second_Delay=5;

    auto Total_Modeling_Time=300;

    // first t can be served only if second facility is seized

    pfacility First_Facility;
    pfacility Second_Facility;

    initlist(Total_Modeling_Time);
    initcreate(1, 0);

    newfac(First_Facility, "\"First Facility\"");
    newfac(Second_Facility, "\"Second Facility\"");

    while (systime<Total_Modeling_Time) {
        plan();
        switch (sysevent) {
            case 1: create(First_Interval); break;
            case 2: if (Second_Facility->status == facility::seized)
seize(First_Facility); else next(7); break;
            case 3: delayt(First_Delay); break;
            case 4: outfac(First_Facility); break;
            case 6: destroy(); break;

            case 7: create(Second_Interval); break;
            case 8: seize(Second_Facility); break;
            case 9: delayt(Second_Delay); break;
            case 10: outfac(Second_Facility); break;
            case 11: destroy(); break;
        }
    }

    cout << "Modeling finished, praise Jesus Christ our Holy Lord GOD
Almighty" << endl << endl;
    printall();
}

int main()
{
    one();
    two();
}

```

```
three();  
return 0;  
}
```

Виконання програми

CIM-CI++ v1.2

Ж. К. Камінська

С. М. Сердюк

НУ"ЗП"

2024

Загальні параметри середовища:										
Поточний час	2424.000									
Поточна подія	5									
Поточний транзакт	1									
Усього подій	761.000									
Час моделювання	0.00 сек.									
Середній час виконання події	0.00000 сек/подія									

ПОДІЯ	1	2	3	4	5	6	7	8	9	10
УСЬОГО	64	64	64	64	64	63	63	63	63	63

ПОДІЯ	11	12	13	14	15	16	17	18	19	20
УСЬОГО	63	63								

Черги

Черга	Кількість входжень	Макс. довжина	Середній час очікування	Середня довжина	% входжень у порожню чергу
	З нульовим часом очікування	Поточна довжина	Без урахування нульових входжень		
"Assembly Queue"	64	1	0.000	0.000	100.000
	64	0	0.000		
	63	1	0.000		
"Firing Queue"	63	0	0.000	0.000	100.000

Накопичувачі

Накопичувач	Ємність	Завантаженість	Середній час перебування	Вміст			Кількість входжень
				Поточн.	Макс.	Середн.	
"Assembly Storage"	3	0.264	30.000	0	1	0.79	64
"Firing Facility"	1	0.211	8.000	0	1	0.21	63

Рисунок 1.1 – Загальне завдання – браузер

```
Enter a name for the Report HTML file: gen
Workers: 4 Profit: -8370
Workers: 5 Profit: -11000
Workers: 6 Profit: -13000

Best Count: 6 Best Result: -13000
```

Enter a name for the Report HTML file: gen
First Queue Length: 0
Second Queue Length: 0
First Queue Length: 0
Second Queue Length: 0
Second Queue Length: 0
First Queue Length: 0
Second Queue Length: 0
Second Queue Length: 1
Second Queue Length: 2
First Queue Length: 0
Second Queue Length: 2
Second Queue Length: 3
Second Queue Length: 4
First Queue Length: 0
Second Queue Length: 4
Second Queue Length: 5
Second Queue Length: 6
First Queue Length: 1
Second Queue Length: 6
Second Queue Length: 7
Second Queue Length: 8
First Queue Length: 1
Second Queue Length: 8
Second Queue Length: 9
Second Queue Length: 10
First Queue Length: 2
Second Queue Length: 10
Second Queue Length: 11
Second Queue Length: 12
First Queue Length: 3
Second Queue Length: 12
Second Queue Length: 13
Second Queue Length: 14
First Queue Length: 3
Second Queue Length: 14
Second Queue Length: 15
Second Queue Length: 16
First Queue Length: 4
Second Queue Length: 16
Second Queue Length: 17
Second Queue Length: 18
First Queue Length: 5
Second Queue Length: 18
Second Queue Length: 19
Second Queue Length: 20
First Queue Length: 5

Second Queue Length: 20
Second Queue Length: 21
Second Queue Length: 22
First Queue Length: 6
Second Queue Length: 22
Second Queue Length: 23
Second Queue Length: 24
First Queue Length: 7
Second Queue Length: 24
Second Queue Length: 25
Second Queue Length: 26
First Queue Length: 7
Second Queue Length: 26
Second Queue Length: 27
Second Queue Length: 28
First Queue Length: 8
Second Queue Length: 28
Second Queue Length: 29
Second Queue Length: 30
First Queue Length: 9
Second Queue Length: 30
Second Queue Length: 31
Second Queue Length: 32
First Queue Length: 9
Second Queue Length: 32
Second Queue Length: 33
Second Queue Length: 34
First Queue Length: 10
Second Queue Length: 34
Second Queue Length: 35
Second Queue Length: 36
First Queue Length: 11
Second Queue Length: 36
Second Queue Length: 37
Second Queue Length: 38
First Queue Length: 11
Second Queue Length: 38
Second Queue Length: 39
Second Queue Length: 40
First Queue Length: 12
Second Queue Length: 40
Second Queue Length: 41
Second Queue Length: 42
First Queue Length: 13
Second Queue Length: 42
Second Queue Length: 43
Second Queue Length: 44
First Queue Length: 13
Second Queue Length: 43
Second Queue Length: 44
Second Queue Length: 45
First Queue Length: 14
Second Queue Length: 45
Second Queue Length: 46
Second Queue Length: 47
First Queue Length: 15
Second Queue Length: 47
Second Queue Length: 48
Second Queue Length: 49
First Queue Length: 15
Second Queue Length: 49
Second Queue Length: 50
Second Queue Length: 51
First Queue Length: 16
Second Queue Length: 51
Second Queue Length: 52
Second Queue Length: 53
First Queue Length: 17
Second Queue Length: 53
Second Queue Length: 54
Second Queue Length: 55
First Queue Length: 17
Second Queue Length: 55
Second Queue Length: 56
Second Queue Length: 57
First Queue Length: 18
Second Queue Length: 57
Second Queue Length: 58
Second Queue Length: 59

First Queue Max Length: 19 Average Length: 7.72768

```
Enter a name for the Report HTML file: gen
Modeling finished, praise Jesus Christ our Holy Lord GOD Almighty
```

Рисунок 1.2 – Загальне завдання – консолька †

Висновки

Таким чином, ми вивчили методи моделювання СМО зі зворотнім зв'язком, а також багатоканальні СМО на основі використання SIMC

Контрольні питання

**Багатоканальні прилади. Множинні типи даних
"НАКОПИЧУВАЧ" (багатоканальний прилад)**

Накопичувач то є багатоканальний прилад де сам накопичувач може бути звільнений не тим транзактом, яким був зайнятий. То є динамічний об'єкт для моделювання декілької пристроїв.

Процедури створення – знищення накопичувача

Створення накопичувачів виконується функцією newstorage, а знищення виконується функцією destrs.

Блокування транзактів

Блокування транзактів здійснюється функцією void wait(event e) де e то є номер очікуваної події.