



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления» (ИУ)  
КАФЕДРА «Информационная безопасность» (ИУ8)

Лабораторная работа №7  
ПО КУРСУ  
«Потоковая многозадачность»

Студент

ИУ8-21  
(Группа)

Г. А. Карев  
(И. О. Фамилия)

Преподаватель:

В. В. Соборова  
(И.О. Фамилия)

## **Цель работы:**

Изучить что такое потоковая многозадачность и написать программы, используя разные методы.

## **Вариант 12 (2 для ЛР7)**

### **Задача:**

Реализовать программу, в которой кроме главного создается три отдельных потока: первый поток печатает  $n_1$  раз результат операций умножения двух вещественных чисел (печатаются 2 операнда и результат операции); второй поток печатает  $n_2$  раз результат операций сложения двух вещественных чисел (печатаются 2 операнда и результат операции), третий поток печатает  $n_3$  значений, выдаваемых ГПСЧ. Значения  $n_1$ ,  $n_2$ ,  $n_3$  задаются константами или вводятся с клавиатуры (примерный диапазон 20..30). Каждый поток имеет свое имя (например, `thread1`, `thread2`, `thread3`), перед печатью результата операции или значением поток должен напечатать с новой строки свое имя. После завершения дочерних потоков главный поток выдает сообщение об окончании работы. Имена потоков и значения  $n_1$ ,  $n_2$ ,  $n_3$  передаются в потоковую функцию через ее параметры. Значения вещественных чисел, над которыми выполняется операция в потоке, передаются в виде 2-х массив в потоковую функцию (операция выполняется над  $i$ -м элементом одного массива и  $i$ -м элементом другого массива). Массивы предварительно до запуска потоков заполняются числами с использованием ГПСЧ.

1. Запустить программу несколько раз при одних и тех же исходных данных, посмотреть, как меняются результаты вывода. Сделать выводы.

2. Между печатью имени потока и результатом операции (значением) установить небольшую задержку, например, 10 мс. Посмотреть, как меняются результаты вывода. Сделать выводы.

### **Часть 2**

Для своего варианта обеспечить синхронизацию потоков:

1. Обеспечить печать имени потока и значения в одну строку без возможных разрывов, продемонстрировать два варианта реализации: использование `mutex` и использование блокировки.

2. С помощью условной переменной обеспечить, чтобы главный поток дожидался завершения дочерних потоков (дочерние потоки перед завершением оповещают главный поток, главный поток принимает эти оповещения). Главный поток после приема оповещения от каждого дочернего потока печатает об этом событии сообщение.

Код:

## ЧАСТЬ 1

*с использованием thread*

---

```
#include <iostream>
#include <thread>
using namespace std;

void thread_operation(const string& name, double* num1, double* num2, int size, char
sign) {
    for (int i = 0; i < size; i++) {
        cout << name << '\n';
        this_thread::sleep_for(chrono::milliseconds(100));

        switch (sign) {
            case '+':
                cout << num1[i] << " + " << num2[i] << " = " << num1[i] + num2[i]
<< '\n';
                break;

            case '*':
                cout << num1[i] << " * " << num2[i] << " = " << num1[i] * num2[i]
<< '\n';
                break;

            case 'r':
                cout << rand() % 1000 / 100.0 << '\n';
                break;
        }
    }
}

int main() {
    srand(time(NULL));
    setlocale(LC_ALL, "RUS");

    int n1, n2, n3;
    cout << "Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во
чисел\n";
    cin >> n1 >> n2 >> n3;
    double* num11 = new double[n1];
    double* num12 = new double[n1];
    double* num21 = new double[n2];
    double* num22 = new double[n2];

    for (int i = 0; i < n1; i++) {
        num11[i] = rand() % 1000 / 100.0;
        num12[i] = rand() % 1000 / 100.0;
    }
    for (int i = 0; i < n2; i++) {
        num21[i] = rand() % 1000 / 100.0;
        num22[i] = rand() % 1000 / 100.0;
    }

    thread t1(thread_operation, "Поток1", num11, num12, n1, '*');
    thread t2(thread_operation, "Поток2", num21, num22, n2, '+');
    t1.join();
    cout << "\nПоток1 завершен\n" << endl;
    t2.join();
    cout << "\nПоток2 завершен\n" << endl;

    thread t3(thread_operation, "Поток3", num11, num12, n3, 'r');
    t3.join();
    cout << "\nПоток3 завершен\n" << endl;

    cout << "\nВсе потоки завершены\n";
    return 0;
}
```

## Вывод программы

Без задержки:

```
Консоль отладки Microsoft Vi x + v
Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во чисел
3 3 3
Поток1
Поток2
5.6 + 7.699.12 * 9.9 = 90.288
Поток1
7.49 + 9.8 = 73.402 =
Поток1
13.293.06
Поток2 * 8.63 = 26.4078
3.43 + 4.82 = 8.25

Поток2
1.56 + 7.76
Поток1 завершен
= 9.32

Поток2 завершен

Поток3
0.41
Поток3
4.67
Поток3
3.34

Поток3 завершен

Все потоки завершены

C:\Users\Admin\source\repos\zxc\x64\Debug\zxc.exe (процесс 13880) завершил
Нажмите любую клавишу, чтобы закрыть это окно:|
```

```
Консоль отладки Microsoft Vi x + v
Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во чисел
3 3 3
Поток2
Поток15.48
+ 0.59 * 1.141.48 = = 0.87326.62
Поток1
Поток2
7 + 7.31 = 14.31

Поток222.05 *
4.855.01 = + 7.589.9425
Поток1
= 12.59
9.41 * 9.58 = 90.1478

Поток1 завершен

Поток2 завершен

Поток3
0.41
Поток3
4.67
Поток3
3.34

Поток3 завершен

Все потоки завершены

C:\Users\Admin\source\repos\zxc\x64\Debug\zxc.exe (процесс 18000) завершил
Нажмите любую клавишу, чтобы закрыть это окно:|
```

С задержкой:

```
Консоль отладки Microsoft Vi x + v
Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во чисел
3 3 3
Поток1
Поток2
2.712.5 + 4.49 * 6.67 = 18.0757
Поток1
= 6.99
Поток2
7.28 * 7.37 = 53.65366.1 + 3.23
= 9.33Поток1
Поток2

6.55 + 6.21 = 0.29 * 0.5112.76
= 0.1479

Поток1 завершен

Поток2 завершен

Поток3
0.41
Поток3
4.67
Поток3
3.34

Поток3 завершен

Все потоки завершены

C:\Users\Admin\source\repos\zxc\x64\Debug\zxc.exe (процесс 4664) завершил
Нажмите любую клавишу, чтобы закрыть это окно:|
```

```
Консоль отладки Microsoft Vi x + v
Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во чисел
3 3 3
Поток1
Поток2
3.69 * 3.19 + 1.67 = 4.864.39 = 16.1991

Поток2Поток1
3.65 * 4.37 + 7.217.4 = = 11.5827.01
Поток1

Поток2
2.17 + 7.869.43 = * 11.65.49
= 43.1514

Поток1 завершен

Поток2 завершен

Поток3
0.41
Поток3
4.67
Поток3
3.34

Поток3 завершен

Все потоки завершены

C:\Users\Admin\source\repos\zxc\x64\Debug\zxc.exe (процесс 7052) завершил
Нажмите любую клавишу, чтобы закрыть это окно:|
```

```
#include <iostream>
#include <future>
using namespace std;

void thread_operation(const string& name, double* num1, double* num2, int size, char
sign) {
    for (int i = 0; i < size; i++) {
        cout << name << "\n";
        this_thread::sleep_for(chrono::milliseconds(100));
        switch (sign) {
            case '+':
                cout << num1[i] << " + " << num2[i] << " = " << num1[i] + num2[i]
<< '\n';
                break;

            case '*':
                cout << num1[i] << " * " << num2[i] << " = " << num1[i] * num2[i]
<< '\n';
                break;

            case 'r':
                cout << rand() % 1000 / 100.0 << '\n';
                break;
        }
    }
}

int main() {
    srand(time(NULL));
    setlocale(LC_ALL, "RUS");

    int n1, n2, n3;
    cout << "Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во
чисел \n";
    cin >> n1 >> n2 >> n3;
    double* num11 = new double[n1];
    double* num12 = new double[n1];
    double* num21 = new double[n2];
    double* num22 = new double[n2];

    for (int i = 0; i < n1; i++) {
        num11[i] = rand() % 1000 / 100.0;
        num12[i] = rand() % 1000 / 100.0;
    }
    for (int i = 0; i < n2; i++) {
        num21[i] = rand() % 1000 / 100.0;
        num22[i] = rand() % 1000 / 100.0;
    }

    future<void> thread1 = async(thread_operation, "Поток1", num11, num12, n1, '*');
    future<void> thread2 = async(thread_operation, "Поток2", num21, num22, n2, '+');
    thread1.get();
    cout << "\nПоток1 завершен\n" << endl;
    thread2.get();
    cout << "\nПоток2 завершен\n" << endl;

    future<void> thread3 = async(thread_operation, "Поток3", num11, num12, n3, 'r');
    thread3.get();
    cout << "\nПоток3 завершен\n" << endl;

    cout << "\nВсе потоки завершены\n";
    return 0;
}
```

## Вывод программы

```
Консоль отладки Microsoft Vi  X  +  v

Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во чисел
3 3 3
Поток1
Поток2
5.36 * 5.52 = 2.14 + 29.58727.16
= Поток19.3

Поток2
3.28 + 8.499.34 = * 11.77
Поток27.85 =
73.319
Поток1
5.41 * 2.046.74 = 11.0364
+
Поток1 завершен
1.89
= 8.63

Поток2 завершен

Поток3
0.41
Поток3
4.67
Поток3
3.34

Поток3 завершен

Все потоки завершены

C:\Users\Admin\source\repos\test\x64\Debug\test.exe (процесс 18920) завершил
Нажмите любую клавишу, чтобы закрыть это окно:
```

## ЧАСТЬ 2

*Избавились от разрывов с использованием mutex*

---

```
#include <iostream>
#include <thread>
#include <condition_variable>
using namespace std;

mutex mute;
condition_variable con_var;
bool flag = false;
bool func = false;
void thread_operation(const string& name, double* num1, double* num2, int size, char
sign) {
    unique_lock<mutex> lock(mute);
    while (!flag) { con_var.wait(lock); }
    flag = false;
    for (int i = 0; i < size; i++) {
        cout << name << '\n';
        this_thread::sleep_for(chrono::milliseconds(100));
        switch (sign) {
            case '+':
                cout << num1[i] << " + " << num2[i] << " = " << num1[i] +
                    num2[i] << '\n';
                break;

            case '*':
                cout << num1[i] << " * " << num2[i] << " = " << num1[i] *
                    num2[i] << '\n';
                break;

            case 'r':
                cout << rand() % 1000 / 100.0 << '\n';
                break;
        }
    }
    func = true;
    con_var.notify_one();
}

int main() {
    setlocale(LC_ALL, "RUS");
    srand(time(NULL));

    int n1, n2, n3;
    cout << "Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во
чисел \n";
    cin >> n1 >> n2 >> n3;
    double* num11 = new double[n1];
    double* num12 = new double[n1];
    double* num21 = new double[n2];
    double* num22 = new double[n2];

    for (int i = 0; i < n1; i++) {
        num11[i] = rand() % 1000 / 100.0;
        num12[i] = rand() % 1000 / 100.0;
    }
    for (int i = 0; i < n2; i++) {
        num21[i] = rand() % 1000 / 100.0;
        num22[i] = rand() % 1000 / 100.0;
    }

    thread t1(thread_operation, "Поток1", num11, num12, n1, '*');
    {
        lock_guard<mutex> lock(mute);
        flag = true;
    }
    con_var.notify_one();
}
```

```

        unique_lock<mutex> lock(mute);
        while (!func) { con_var.wait(lock); }
        func = false;
        cout << "Поток1 завершен\n\n";
    }

    thread t2(thread_operation, "Поток2", num21, num22, n2, '+');
    {
        lock_guard<mutex> lock(mute);
        flag = true;
    }
    con_var.notify_one();
    {
        unique_lock<mutex> lock(mute);
        while (!func) { con_var.wait(lock); }
        func = false;
        cout << "Поток2 завершен\n\n";
    }

    thread t3(thread_operation, "Поток3", num11, num12, n3, 'r');
    {
        lock_guard<mutex> lock(mute);
        flag = true;
    }
    con_var.notify_one();
    {
        unique_lock<mutex> lock(mute);
        while (!func) { con_var.wait(lock); }
        func = false;
        cout << "Поток3 завершен\n";
    }

    t1.join();
    t2.join();
    t3.join();
    cout << "\nВсе потоки завершены\n";
    return 0;
}

```



## Вывод программы

```
Консоль отладки Microsoft Vi  ×  +  ∨  
Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во чисел  
3 3 3  
Поток1  
5.23 * 4.22 = 22.0706  
Поток1  
5.4 * 7.15 = 38.61  
Поток1  
3.65 * 0.1 = 0.365  
Поток1 завершен  
  
Поток2  
3.66 + 9.4 + 13.06  
Поток2  
6.68 + 7.11 + 13.79  
Поток2  
3.25 + 6.04 + 9.29  
Поток2 завершен  
  
Поток3  
0.41  
Поток3  
4.67  
Поток3  
3.34  
Поток3 завершен  
  
Все потоки завершены  
  
C:\Users\Admin\source\repos\test\x64\Debug\test.exe (процесс 8312) завершил  
Нажмите любую клавишу, чтобы закрыть это окно:
```

```
#include <iostream>
#include <mutex>
#include <thread>
#include <future>
using namespace std;

mutex mute, mute1;
void thread_operation(const string& name, double* num1, double* num2, int size, char
sign) {
    mute.lock();
    for (int i = 0; i < size; i++) {
        cout << name << '\n';
        this_thread::sleep_for(chrono::milliseconds(100));
        switch (sign) {
            case '+':
                cout << num1[i] << " + " << num2[i] << " = " << num1[i] + num2[i]
<< '\n';
                break;

            case '*':
                cout << num1[i] << " * " << num2[i] << " = " << num1[i] * num2[i]
<< '\n';
                break;

            case 'r':
                cout << rand() % 1000 / 100.0 << '\n';
                break;
        }
    }
    mute.unlock();
}

int main() {
    setlocale(LC_ALL, "RUS");
    srand(time(NULL));

    int n1, n2, n3;
    cout << "Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во
чисел\n";
    cin >> n1 >> n2 >> n3;
    double* num11 = new double[n1];
    double* num12 = new double[n1];
    double* num21 = new double[n2];
    double* num22 = new double[n2];

    for (int i = 0; i < n1; i++) {
        num11[i] = rand() % 1000 / 100.0;
        num12[i] = rand() % 1000 / 100.0;
    }
    for (int i = 0; i < n2; i++) {
        num21[i] = rand() % 1000 / 100.0;
        num22[i] = rand() % 1000 / 100.0;
    }

    future<void> thread1 = async(thread_operation, "Поток1", num11, num12, n1, '*');
    future<void> thread2 = async(thread_operation, "Поток2", num21, num22, n2, '+');
    future<void> thread3 = async(thread_operation, "Поток3", nullptr, nullptr, n3,
'r');

    thread1.get();
    thread2.get();
    thread3.get();

    cout << "Все потоки завершены\n";
    return 0;
}
```

## Вывод программы

---

```
Введите 1)Кол-во операций умножения, 2)Операций сложения, 3)Кол-во чисел
3 3 3
Поток1
6.92 * 5.15 = 35.638
Поток1
9.73 * 8.23 = 80.0779
Поток1
6.67 * 5.92 = 39.4864
Поток2
3.74 + 5.62 = 9.36
Поток2
3.03 + 6.92 = 9.95
Поток2
6.34 + 8.94 = 15.28
Поток3
0.41
Поток3
4.67
Поток3
3.34
Все потоки завершены

C:\Users\Admin\source\repos\test\x64\Debug\test.exe (процесс 4672) завершил
Нажмите любую клавишу, чтобы закрыть это окно:
```

### Вывод:

Научился использовать потоковую многозадачность, написал работоспособный код и выполнил задачу.