

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ОТЧЁТ
по лабораторной работе №5
“Многопоточность.”

Выполнил:
студент группы ПО-9
Тусюк Т.В.

Проверил:
Козик И.Д.

Брест 2023

Лабораторная работа №5

Вариант 22

Цель работы: научиться работать с многопоточностью в приложениях.

Основное содержание работы.

Необходимо написать 2 программы, используя в них несколько потоков. Одну из программ реализовать через атомные переменные, вторую – через mutex. Отчёт должен содержать цель работы, код программы, скриншот работы программы и вывод по лабораторной работе.

Задание №3: создать два потока, последовательно увеличивающих и уменьшающих переменную заданное с клавиатуры количество раз на заданные с клавиатуры значения.

Текст программы:

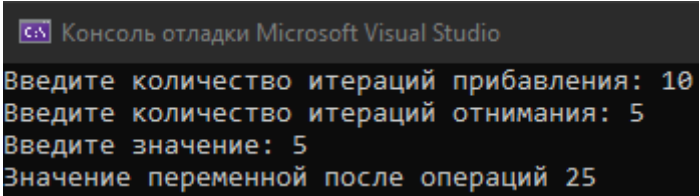
Программа с использованием атомной переменной:

```
#include <iostream>
#include <thread>
#include <atomic>
#include <mutex>
using namespace std;
int main()
{
    setlocale(LC_ALL, "ru");

    int value, iterationsPlus, iterationsMinus;
    std::cout << "Введите количество итераций прибавления: ";
    std::cin >> iterationsPlus;
    std::cout << "Введите количество итераций отнимания: ";
    std::cin >> iterationsMinus;
    std::cout << "Введите значение: ";
    std::cin >> value;

    atomic<float> counter(0);
    thread thread1([&]() {
        for (int i = 0; i < iterationsPlus; ++i) {
            counter.store(counter.load() + value);
        }
    });
    thread thread2([&]() {
        for (int i = 0; i < iterationsMinus; ++i) {
            counter.store(counter.load() - value);
        }
    });
    thread1.join();
    thread2.join();
    cout << "Значение переменной после операций " << counter.load();
    return 0;
}
```

Результат программы:



Консоль отладки Microsoft Visual Studio

Введите количество итераций прибавления: 10
Введите количество итераций отнимания: 5
Введите значение: 5
Значение переменной после операций 25

Программа с использованием мьютексов:

```
#include <iostream>
#include <mutex>
#include <thread>

using namespace std;

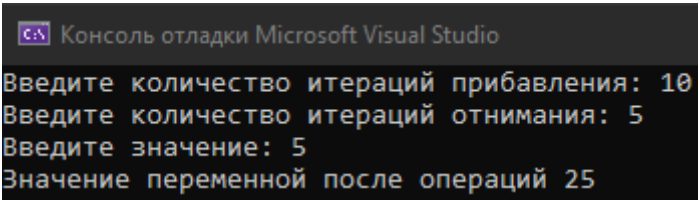
int main() {
    setlocale(LC_ALL, "ru");
    int value, iterationsPlus, iterationsMinus;
    std::cout << "Введите количество итераций прибавления: ";
    std::cin >> iterationsPlus;
    std::cout << "Введите количество итераций отнимания: ";
    std::cin >> iterationsMinus;
    std::cout << "Введите значение: ";
    std::cin >> value;

    float counter = 0;
    mutex counterMutex;
    thread thread1([&](){
        for (int i = 0; i < iterationsPlus; ++i) {
            counterMutex.lock();
            counter += value;
            counterMutex.unlock();
        }
    });
    thread thread2([&](){
        for (int i = 0; i < iterationsMinus; ++i) {
            counterMutex.lock();
            counter -= value;
            counterMutex.unlock();
        }
    });
    thread1.join();
    thread2.join();

    cout << "Значение переменной после операций " << counter << endl;

    return 0;
}
```

Результат программы:



Консоль отладки Microsoft Visual Studio

```
Введите количество итераций прибавления: 10
Введите количество итераций отнимания: 5
Введите значение: 5
Значение переменной после операций 25
```

Вывод: Научился работать с многопоточностью в приложениях.