

Министерство цифрового развития, связи
и массовых коммуникаций Российской Федерации

Сибирский государственный университет
телекоммуникаций и информатики

Кафедра прикладной математики и кибернетики

ЛАБОРАТОРНАЯ РАБОТА №9

По дисциплине: «Операционные системы»

Выполнили:

Студенты 3 курса группы ИП-111
Корнилов А.А.,
Попов М.И.,
Толкач А.А.

Проверил:

Профессор кафедры ПМиК
Малков Е.А.

Новосибирск, 2023

Задание: протестируйте спин-блокировку используя фрагменты кода лекции 9.

Цель: протестируйте спин-блокировку используя фрагменты кода лекции 9

Выполнение работы:

Для демонстрации спин-блокировки была выбрана программа 8 лабораторной работы. Спин-блокировка реализована через `atomic_flag lock1`. Блокировка устанавливается `lock1.test_and_set(memory_order_acquire)` и пока значение `false` он заходит в критическую область.

```
#include <iostream>
#include <iomanip>
#include <thread>
#include <vector>
#include <atomic>
#include <mutex>

using namespace std;

const int iterations_per_thread = 125000000; // Количество итераций на каждом потоке

atomic<double> pi{ 0.0 }; // Используем атомарную переменную для суммирования
atomic_flag lock1 = ATOMIC_FLAG_INIT; // Используем atomic_flag для спин-блокировки

void calculatePi(int start, int end) {
    double localSum = 0;
    int j = 2 * start + 1;

    for (int i = start; i < end; i++){
        if (i % 2 == 0){
            localSum += 4.0 / j;
        }
        else {
            localSum -= 4.0 / j;
        }
        j += 2;
    }

    while (lock1.test_and_set(memory_order_acquire)) {} // Захватываем блокировку

    pi += localSum;

    lock1.clear(memory_order_release); // Освобождаем блокировку
}

int main() {
    int num_threads = thread::hardware_concurrency();
    cout << "num_threads = " << num_threads << endl;
    //int num_threads = 1;

    vector<thread> threads;

    for (int i = 0; i < num_threads; i++) {
        int start = i * iterations_per_thread;
        int end = start + iterations_per_thread;
        threads.push_back(thread(calculatePi, start, end));
    }
}
```

```

    }

    for (auto& t : threads) t.join();
    cout << "Реальное число pi:
3.14159265358979323846264338327950288419716939937510582097494459" << endl;
    cout << setprecision(64) << "Вычисляемое число pi: " << pi << endl;

    return 0;
}

```

Листинг 1 – программа lab09_3.c

Программа для компиляции программы и вывод работы:

```

miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/9$ g++ -std=c++20 -pthread lab09_3.cpp
-o lab09_3
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/9$ ./lab09_3
num_threads = 12
Реальное число pi: 3.14159265358979323846264338327950288419716939937510582097494459
Вычисляемое число pi: 3.14159265327111913990165703580714762210845947265625
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/9$ ./lab09_3
num_threads = 12
Реальное число pi: 3.14159265358979323846264338327950288419716939937510582097494459
Вычисляемое число pi: 3.14159265327111913990165703580714762210845947265625
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/9$ ./lab09_3
num_threads = 12
Реальное число pi: 3.14159265358979323846264338327950288419716939937510582097494459
Вычисляемое число pi: 3.14159265327111913990165703580714762210845947265625
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/9$ ./lab09_3
num_threads = 12
Реальное число pi: 3.14159265358979323846264338327950288419716939937510582097494459
Вычисляемое число pi: 3.14159265327111913990165703580714762210845947265625
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B BY3/OS/9$

```