Кафедра прикладной математики и кибернетики

Выполнили:

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

Толкач А.А.

Проверил:

Профессор кафедры ПМиК  
Малков Е.А.

ЛАБОРАТОРНАЯ РАБОТА №9

По дисциплине: «Операционные системы»

Сибирский государственный университет  
телекоммуникаций и информатики

Новосибирск, 2023

Министерство цифрового развития, связи  
и массовых коммуникаций Российской Федерации

**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/В ВУЗ/OS/9$ g++ -std=c++20 -pthread lab09\_3.cpp -o lab09\_3  miron@DESKTOP-UMC1Q46:/mnt/u/Documents/В ВУЗ/OS/9$ ./lab09\_3  num\_threads = 12  Реальное число pi: 3.14159265358979323846264338327950288419716939937510582097494459  Вычисляемое число pi: 3.14159265327111913990165703580714762210845947265625  miron@DESKTOP-UMC1Q46:/mnt/u/Documents/В ВУЗ/OS/9$ ./lab09\_3  num\_threads = 12  Реальное число pi: 3.14159265358979323846264338327950288419716939937510582097494459  Вычисляемое число pi: 3.14159265327111913990165703580714762210845947265625  miron@DESKTOP-UMC1Q46:/mnt/u/Documents/В ВУЗ/OS/9$ ./lab09\_3  num\_threads = 12  Реальное число pi: 3.14159265358979323846264338327950288419716939937510582097494459  Вычисляемое число pi: 3.14159265327111913990165703580714762210845947265625  miron@DESKTOP-UMC1Q46:/mnt/u/Documents/В ВУЗ/OS/9$ ./lab09\_3  num\_threads = 12  Реальное число pi: 3.14159265358979323846264338327950288419716939937510582097494459  Вычисляемое число pi: 3.14159265327111913990165703580714762210845947265625  miron@DESKTOP-UMC1Q46:/mnt/u/Documents/В ВУЗ/OS/9$ ./lab09\_3  num\_threads = 12  Реальное число pi: 3.14159265358979323846264338327950288419716939937510582097494459  Вычисляемое число pi: 3.14159265327111913990165703580714762210845947265625  miron@DESKTOP-UMC1Q46:/mnt/u/Documents/В ВУЗ/OS/9$ |