|  |  |
| --- | --- |
|  | **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 ИУ «Информатика и системы управления»

**КАФЕДРА ИУ8 «Компьютерная безопасность*»***

**ЛАБОРАТОРНАЯ РАБОТА №7**

**Название лабораторной работы: Потоковая многозадачность.**

**по курсу: Алгоритмические языки.**

**Группа ИУ8 – 24**

**Студент: Полонский Сергей**

**Преподаватель: Барыкин Дмитрий**

**Дата: 10.05.2024**

**ОТЧЕТ:**

**1)** **Цель работы:**

Научиться работать с потоковой многозадачностью.

**2)** **Условие задачи:**

## Реализовать программу, в которой кроме главного создается три отдельных потока: первый поток печатает n1 раз результат операций умножения двух вещественных чисел (печатаются 2 операнда и результат операции); второй поток печатает n2 раз результат операций сложения двух вещественных чисел (печатаются 2 операнда и результат операции), третий поток печатает n3 значений, выдаваемых ГПСЧ. Значения n1, n2, n3 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 n1, n2, n3 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.

**3) Текст программы с коментариями:**

#include <iostream>

#include <thread>

#include <ctime>

#include <chrono>

#include <mutex>

using namespace std;

mutex mtx; // переменная класса mytex

void operation(string namethread, const int n, const double\* arr1, const double\* arr2, const char oper) {

mtx.lock(); // с этого момента выполнять функцию может лишь единственный поток одновременно

for (int i = 0; i < n; ++i) {

this\_thread::sleep\_for(chrono::milliseconds(10)); // ожидание 10 милисекунд

if (oper == '\*') {

cout << namethread << " - umnogenie: " << arr1[i] << " \* " << arr2[i] << " = " << arr1[i] \* arr2[i] << endl;

}

else

if (oper == '+') {

cout << namethread << " - slogenie: " << arr1[i] << " + " << arr2[i] << " = " << arr1[i] + arr2[i] << endl;

}

else if (oper == 'r') {

cout << namethread << " - random: " << rand() % 10 + 1 << endl;

}

}

mtx.unlock();

}

int main() {

srand(time(0)); // задание точки в системе для функции рандом

const int n1 = 20;

const int n2 = 25;

const int n3 = 30;

double arr1[n3];

double arr2[n3];

for (int i = 0; i < n1; ++i) {

arr1[i] = rand() % 10 + 1;

arr2[i] = rand() % 10 + 1;

}

thread thread1(operation, "thread1", n1, arr1, arr2, '\*'); // в поток помещаем функцию и аргументы к ней

thread thread2(operation, "thread2", n2, arr1, arr2, '+');

thread thread3(operation, "thread3", n3, nullptr, nullptr, 'r'); // в данном случае arr1 и arr2 использоваться не будут, по этому nullptr

thread1.join(); // ожидаем каждый поток

thread2.join();

thread3.join();

cout << "Finally end!!!" << endl;

}

**4) Вывод:**

я научился работать с потоковой многозадачностью.