МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №1
по дисциплине «Параллельные алгоритмы»
Тема: Основы работы с процессами и потоками.

Студент гр. 9303 _____ Камакин Д.В.

Преподаватель _____ Сергеева Е.И.

Санкт-Петербург
2022

Цель работы.

Ознакомиться с основами работы с процессами и потоками в языке C++.

Задание.

Выполнить поэлементное сложение 2х матриц M*N. Входные матрицы вводятся из файла (или генерируются). Результат записывается в файл

1.

Выполнить задачу, разбив её на 3 процесса. Выбрать механизм обмена данными между процессами.

Процесс 1: заполняет данными входные матрицы (читает из файла или генерирует их некоторым образом).

Опционально: в этом процессе могут быть 2 потока ввода/генерации данных

Процесс 2: выполняет сложение

Процесс 3: выводит результат

2.

Аналогично 1.1, используя потоки (threads)

2.1.

Разбить сложение на P потоков.

Исследовать зависимость между количеством потоков, размерами входных данных и параметрами целевой вычислительной системы.

Выполнение работы.

Задание 1.

Дочерние процессы создаются с помощью fork(), для общения между ними была использована библиотека boost.

Задания 2.

Задание 1 было переписано под использование потоков, а поскольку у них есть общая память, то для общения использовались классы STL, а именно: std::promise и std::future.

Задание 3.

Задание 2 было дополнено параллельным суммированием матрицы. Была добавлена новая входная матрица размером 500x500. Каждому потоку выдавался размер матриц, который требовалось просуммировать. Была исследована зависимость между размером двумерного массива, количеством потоков и временем выполнения программы.

Для начала рассмотрим “небольшую” матрицу размером 100x100:

Количество потоков	Время выполнения(мс)
1	811
2	497
3	551
4	472
5	612
6	538
7	570
8	729
9	700
10	659
11	744
12	725
13	729
14	1249
15	989

Таблица 1 - Исследование скорости выполнения программы от количества потоков для матрицы 100x100

Видно, что самый большой прирост в скорости работы был при 2 потоках. Далее результаты либо близки к этому значению, либо имеют тенденцию на деградацию. Связано это с тем, что переключение контекста и создание новых потоков - дорогая операция, которая зачастую может быть дороже, чем работа в одном потоке.

Теперь рассмотрим матрицу размером 500x500

Количество потоков	Время выполнения(мс)
1	11046
2	5912
3	9350
4	5199
5	5009
6	4991
7	4135
8	4688
9	5064
10	5015
11	4720
12	6838
13	5156
14	4750
15	6380

Таблица 2 - Исследование скорости выполнения программы от количества потоков для матрицы 500x500

Здесь ситуация немного иная: программа выполняется даже на 7 потоках быстрее, чем на 2. Связано это с размером самого массива данных: здесь работа

по распараллеливанию дешевле, чем расчёт информации, что и даёт такой прирост. Стоит отметить, что эти показатели могут сильно разниться и от CPU, и от операционной системы, и даже от внешних условий.

Выводы.

В ходе выполнения лабораторной работы мы ознакомились с основами работы с процессами и потоками в языке C++.