**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Операционные системы»

**Лабораторная работа № 3**

Тема: p\_thread

Студент: Гамов Павел

Группа: 80-207

Преподаватель: Миронов Е.С.

Дата:

Оценка:

Москва, 2019

**1.**               **Постановка задачи**

Найти среднее арифметическое шестнадцатеричных чисел, поданных из файла используя потоки C.

**2.**           **Описание программы**

Аргументами программы является максимально количество потоков, которое можно использовать при выполнении задачи. Вторым аргументов является название файла откуда будут подаваться числа.

Первым делом числа заносятся в очередь, которая представляет собой массив векторов из чаров, тоесть каждая цифра числа хранится в шестнадцатеричном формате в каждой ячейке массива. Таким образом можно хранить огромные числа. Реализуем функции суммы двух чисел в таком представлении. Дальше функцию деления числа на 2.

Основной принцип работы заключается в том, что мы перемещаем вектора в массив структур, которые в дальнейшем будут отправлены в функции потока, реализуем такую функцию. Если тред получил данные, он суммирует два числа, делит их на 2, отправляет обратно.

Когда файл считался, мы в цикле заполняем структуру, считая кол-во используемых тредов, дальше в цикле создаем треды, передавая им структуры. После чего в цикле вызываем функцию join, тем самым завершая функции тредов. Полученные структуры изменены, результат работы добавляются в общую очередь. Так происходит пока размер очереди не станет равна 1. Полученное число пишется в конце работы программы, также пишется время работы в секундах, которое было получено, используя функции таймера из time.h.

**3.**           **Выводы**

Используя мульти поточность можно заметно ускорить программу, если числа реально длинные и большие, то линейное решение задачи представляет собой полный кошмар, так как функции суммы и деления чисел ну уж очень долго делаются. Значит этот фрагмент задачи можно пустить на распараллеливание. Использования такого метода часто сопровождается использованием дополнительной памяти, что не всегда удается использовать на слабых машинах, также кол-во тредов зависит от возможностей ЦП.

Очень понравилась возможность создания тредпула, но из за отсутствия функции joinable динамически выделять и отправлять на проверку данные сложно, все равно приходится отправлять данные пакетами, линейно копируя данные.