|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ |
| **«Национальный исследовательский ядерный университет «МИФИ»**  **(НИЯУ МИФИ)** |
| ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ |
| КАФЕДРА «КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ» (№12) |

Лабораторная работа №3

«Реализация через MPI»

по дисциплине «Гибридные суперкомпьютерные технологии»

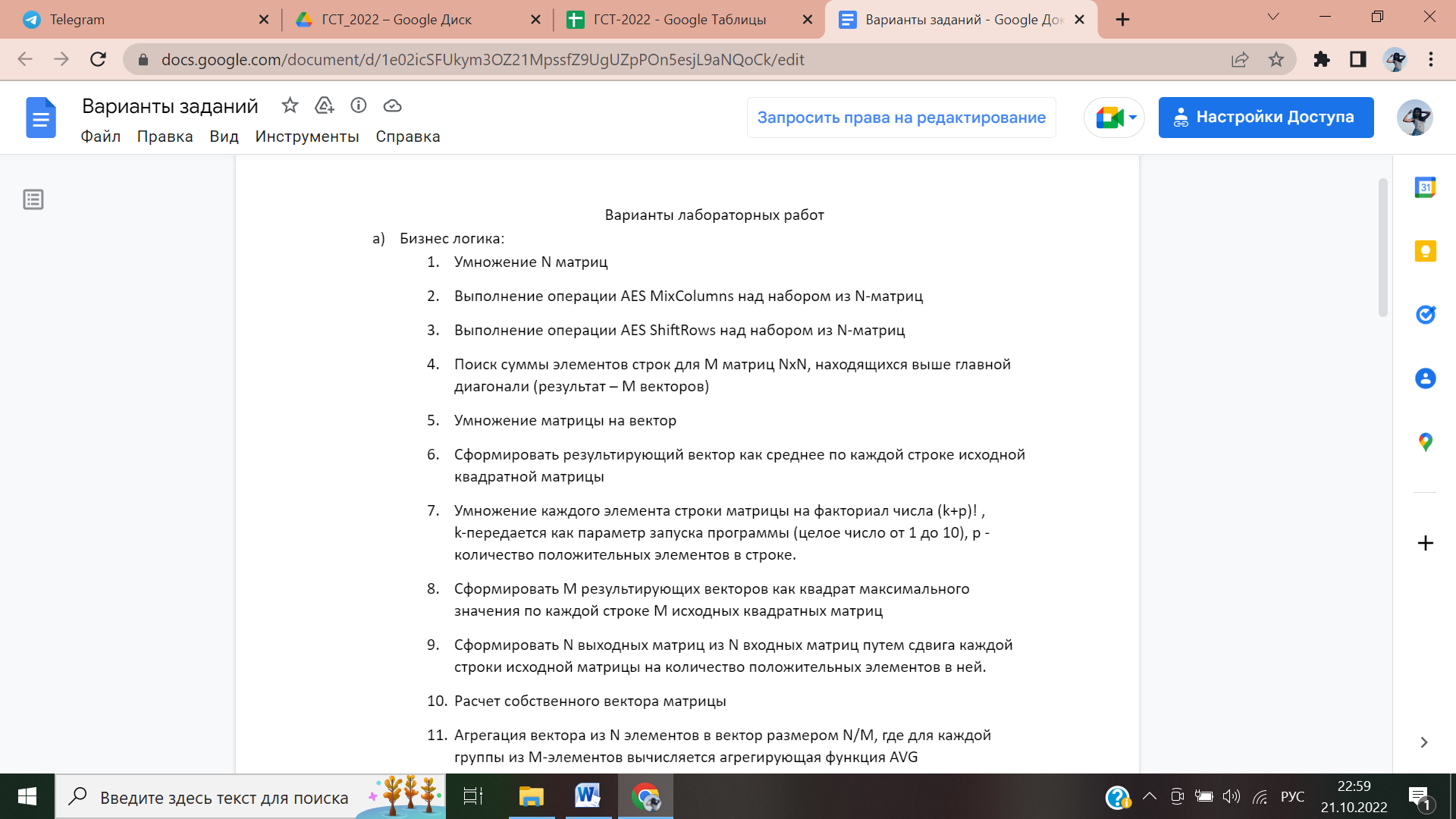
Работу выполнила студентка группы М21-502:

Нургазиева Д.Р.

Проверил: Синельников Д.М.

2022

**Цель:** изучить основы реализация программы через MPI. Выполнить задание по варианту 5:



Особенности исполнения в каждом из вариантов:

* Считывание данных происходит из файла (либо данные передаются по протоколу TCP)
* Данные генерируются утилитой, принимающей в качестве параметров размер данных для обработки в мегабайтах и имя файла (TCP хост-порт) куда будут выгружены данные
* Программа выполняет бизнес-логику и записывает результат в выходной файл (отправляет данные на порт возврата результатов программы-генератора по TCP, сохранение файла с результатами осуществляет программа-генератор)
* В конце файла с результатами сохраняется информация о времени выполнения вычислений и размере обработанных данных

# **Ход работы:**

Программа состоит из двух подпрограмм:

1. Generator.c – отвечает за генерацию данных;
2. Counter.c – выполняет умножение матрицы на вектор, подсчитывает время.

В первую очередь запускается generator.c и вводится размер файла и имя файла. Генерируется матрица с цифрами от 0 до 100. Количество строк и столбцов в зависимости от мегабайт высчитывается по формуле.

По сути, в среднем имеем 3 бита на одно число (пробел + само число в 90: случаев состоящее из 2х цифр). Сгенерированные данные грузятся файл

Далее запускается counter.c и вводится размер файла и имя файла.

Происходит: 1. Выделение памяти под массив, матрицу. 2. перемножение матрицы на вектор, в рамках этого процесса подсчитывается время выполнения. Также происходит реализация через MPI. Приницип работы представлен на рисунке 1.

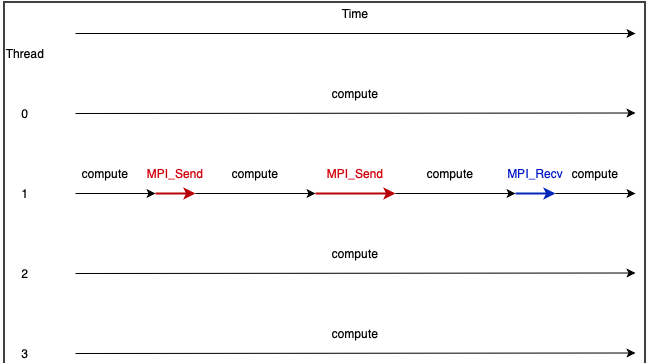


Рисунок 1 – MPI

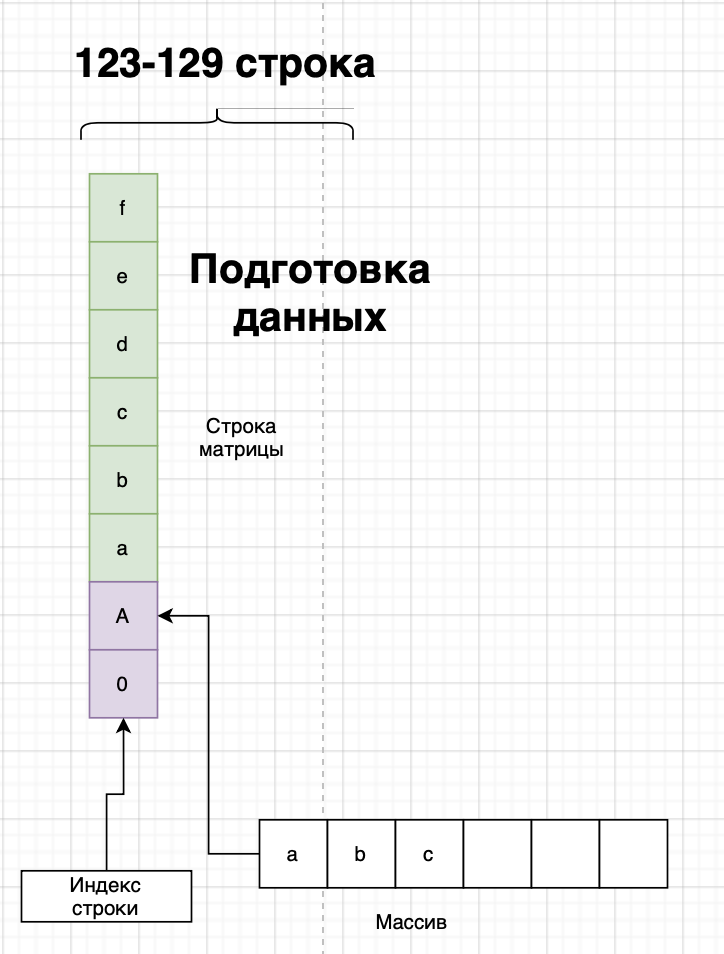


Рисунок 2 – Принцип работы программы

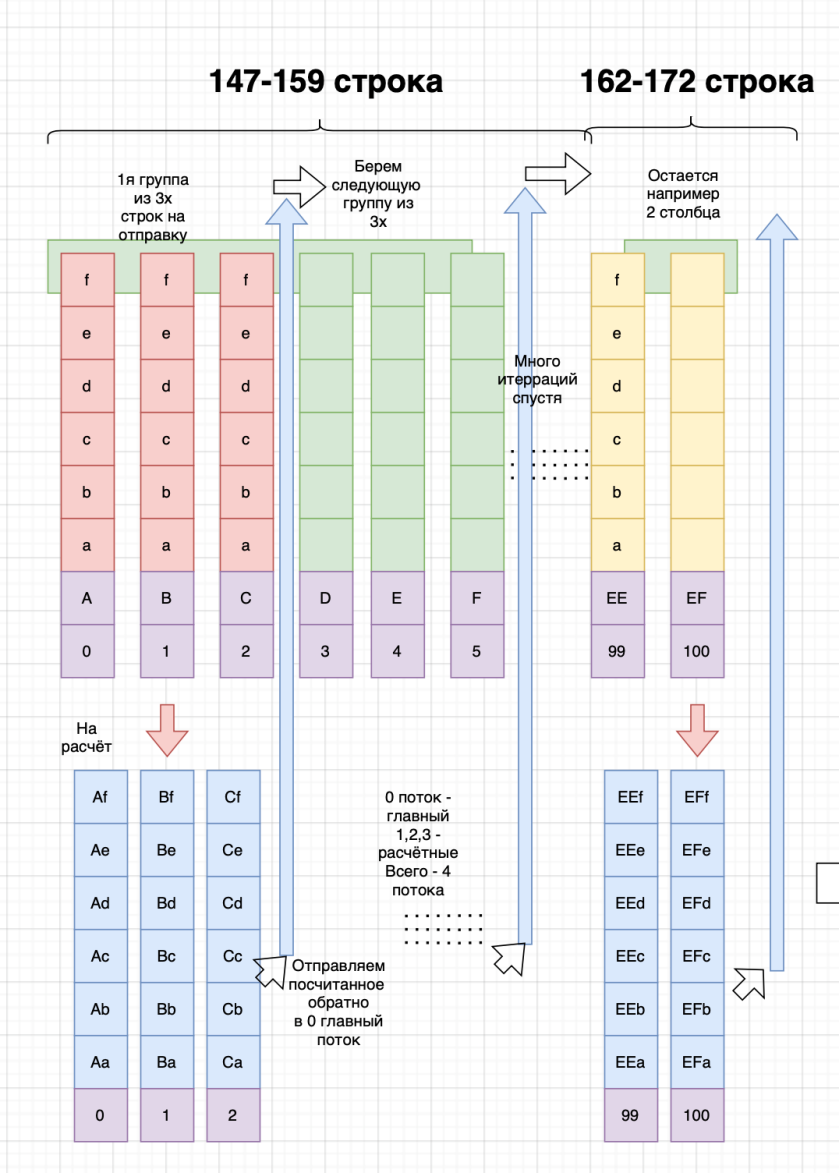


Рисунок 3 - Принцип работы программы

MPI использует объекты называемые коммуникаторы

• Defines which processes can talk

• Communicators have a size

MPI\_COMM\_WORLD

• Predefined as ALL of the MPI Processes

• Size = Nprocs

Rank

• Integer process identifier

• 0≤Rank<Size

MPI\_Comm\_size //количество процессов

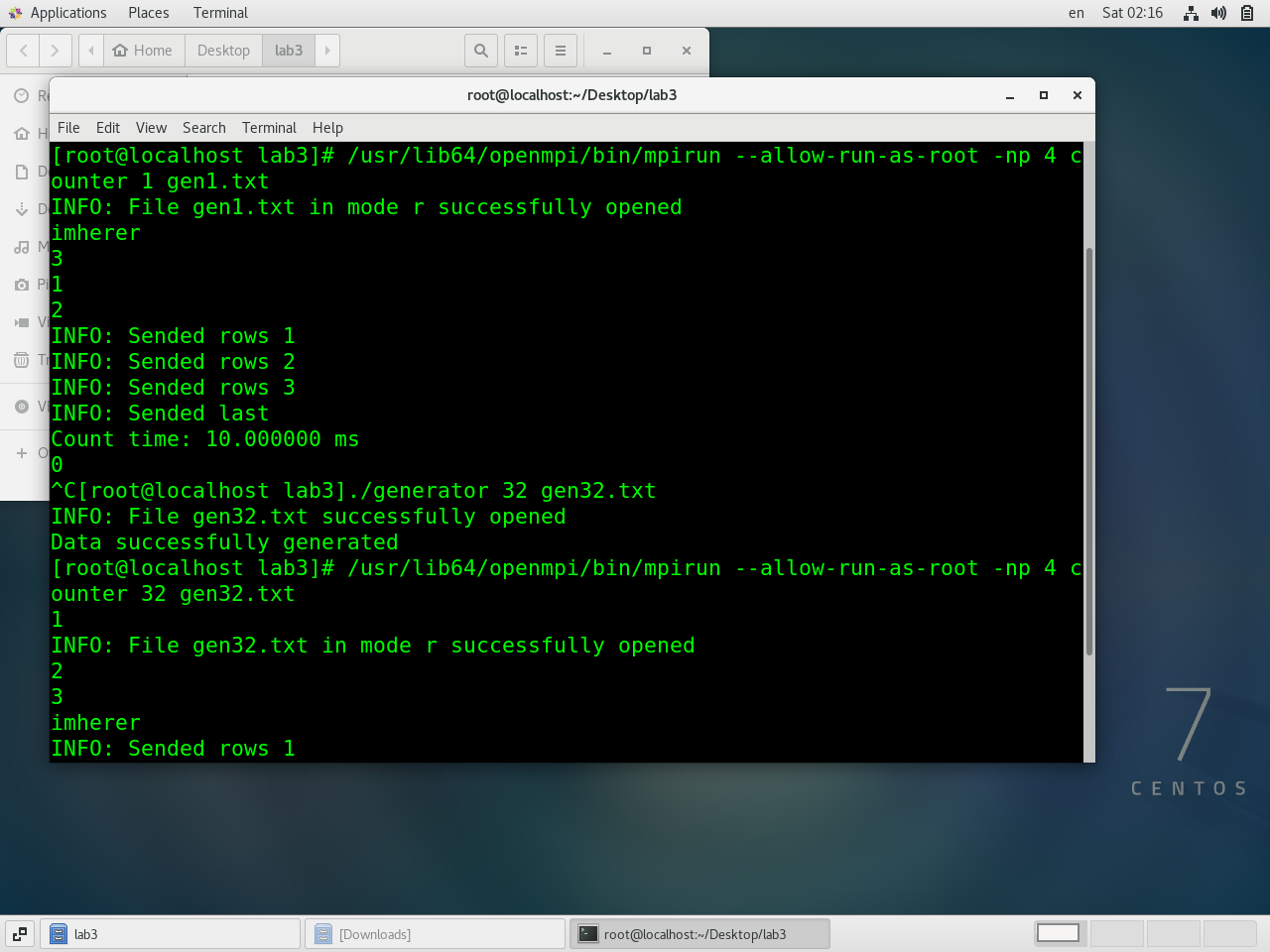
MPI\_Comm\_rank //текущий process id

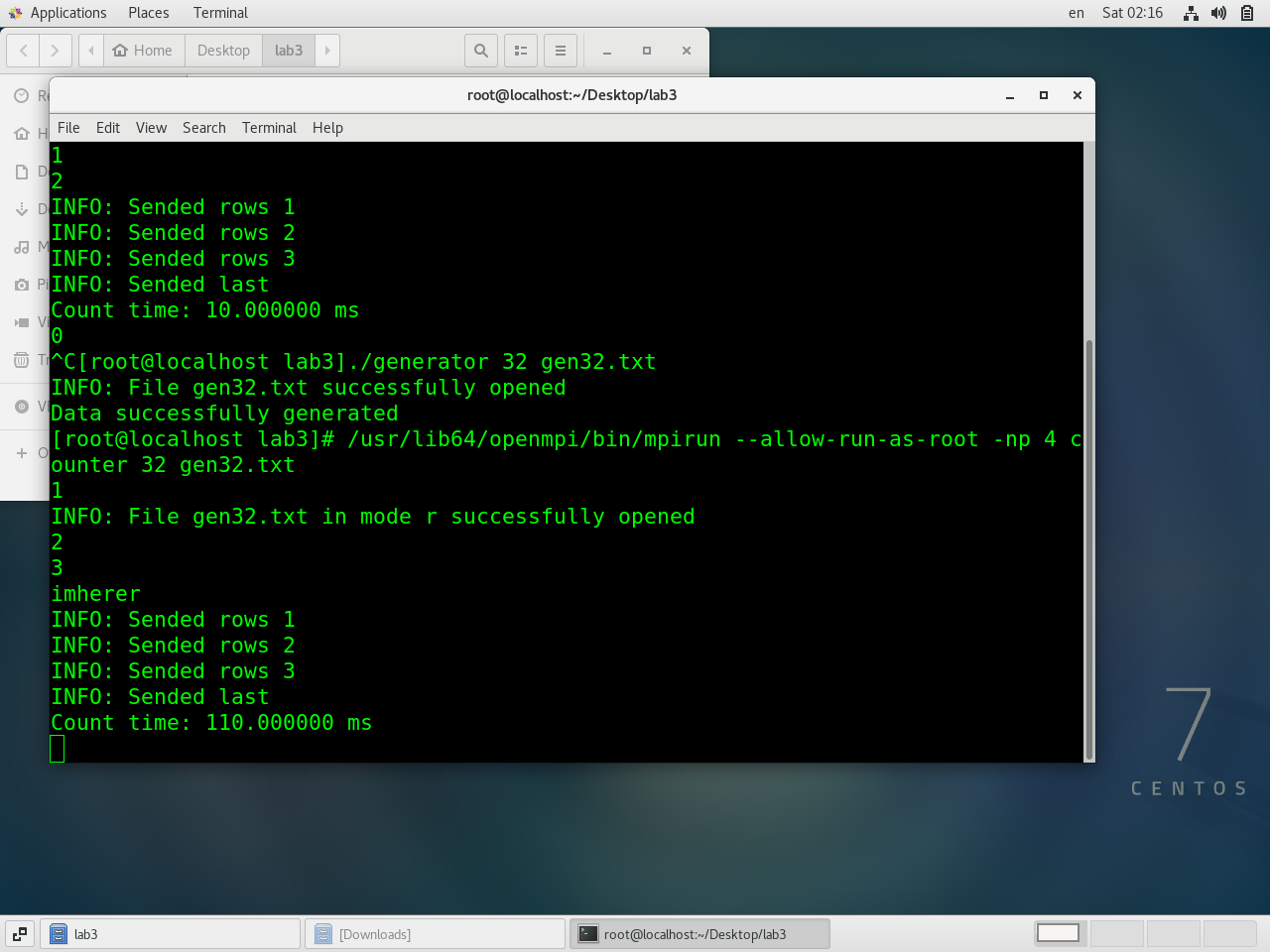
mpicc -o counter ./counter.c

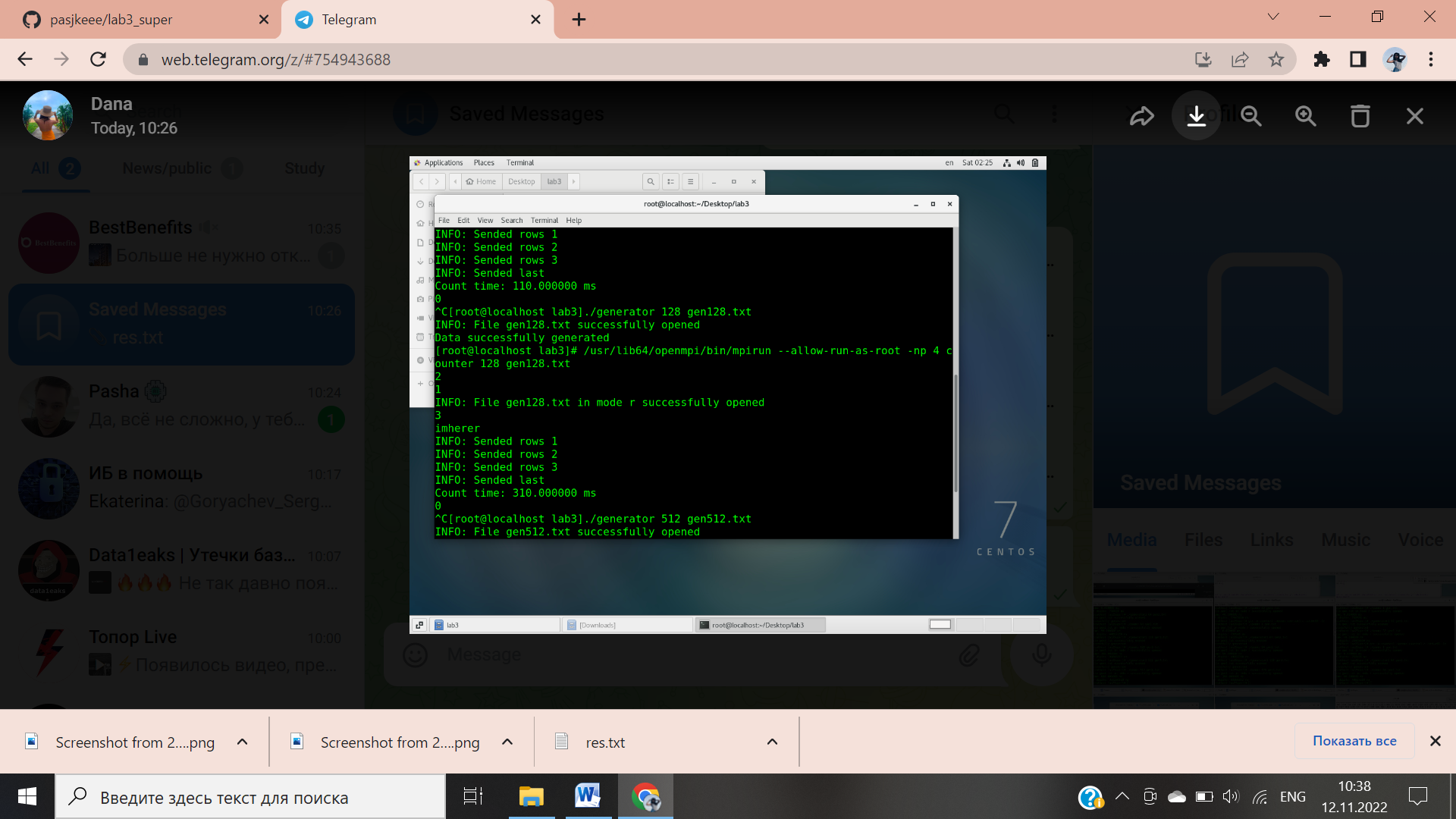
mpirun -np 4 counter.c

Главный поток - 0 он принимает и отправляет данные. Все остальные потоки – вычислительные. На каждый поток подается по строке, считается и отправляется обратно

**Выполнение программы:**

****

****



**Также результирующая матрица выводится в файл. Файл приложен в папке.**

Сравнительная таблица параллельной реализации и реализации через MPI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Метод | Время (мс)на 1мб | Время (мс)на 32мб | Время (мс)на 64мб | Время (мс)на 128мб | Время (мс)на 512мб |
| Параллельный | 10 | 50 | 120 | 260 | 1080 |
| OpenMP | 0 | 40 | 90 | 190 | 860 |
| MPI | 10 | 110 |  | 310 |  |

**Вывод:** были изучены основы реализации MPI. Проанализировано время выполнения программы при разных подходах.