

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет «Информатика и вычислительная техника»

Кафедра «Математика и информатика»

**Лабораторная работ**

**по дисциплине « Параллелное программирование»**

Выполнил:

Студент Гантсу С.Ж.

Группа МСК11

Проверил:

Литвинов В.Н.

Ростов-на-Дону2025 г.

**1. Введение**

В рамках данной работы были реализованы 18 заданий с использованием интерфейса MPI (Message Passing Interface). Задания охватывают базовые и продвинутые техники параллельного программирования, включая:

- Точечные и коллективные коммуникации.

- Динамическое создание процессов.

- Гибридное программирование (MPI + OpenMP).

**Цель работы**: освоение методов параллельных вычислений для эффективного решения вычислительных задач.

2. Обзор реализованных заданий

2.1. Базовые задания (15–22)

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Название** | **Описание** | **Ключевые функции MPI** |
| 15 | "I am!" | Вывод информации о каждом процессе | MPI\_Comm\_rank, MPI\_Comm\_size |
| 16 | "На первый-второй рассчитайся!" | Разделение процессов на четные/нечетные | MPI\_Comm\_rank |
| 17 | Блокирующие обмены | Передача сообщения между процессами | MPI\_Send, MPI\_Recv |
| 18 | "Эстафетная палочка" | Кольцевая передача данных | MPI\_Send, MPI\_Recv |
| 19 | "Master-Slave" | Главный процесс собирает данные от подчиненных | MPI\_Send, MPI\_Recv |
| 20 | Неблокирующие обмены | Асинхронная передача сообщений | MPI\_Isend, MPI\_Irecv, MPI\_Wait |
| 21 | "Сдвиг по кольцу" | Одновременные отправка и прием | MPI\_Sendrecv |
| 22 | "Каждый каждому" | Полносвязный обмен сообщениями | MPI\_Isend, MPI\_Irecv, MPI\_Waitall |

2.2. Коллективные операции (23–26)

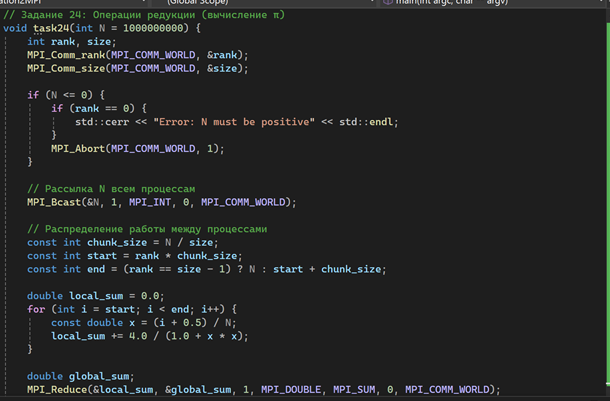
|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Название** | **Описание** | **Ключевые функции MPI** |
| 23 | Широковещательная рассылка | Подсчет частоты символов в строке | MPI\_Bcast, MPI\_Reduce |
| 24 | Операции редукции | Вычисление числа π методом Монте-Карло | MPI\_Bcast, MPI\_Reduce |
| 25 | Распределение данных | Умножение матриц с распределением работы | MPI\_Scatter, MPI\_Gather |
| 26 | Группы и коммуникаторы | Создание подгрупп процессов | MPI\_Comm\_group, MPI\_Comm\_create |

2.3. Продвинутые задания (27–32)

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **№** | **Название** | **Описание** | **Ключевые технологии** |
| 27 | Динамические процессы | Создание новых процессов во время выполнения | MPI\_Comm\_spawn |
| 28 | Односторонние коммуникации | Вычисление π с использованием RMA | MPI\_Win\_create, MPI\_Accumulate |
| 29 | Исследование масштабируемости | Анализ времени выполнения при разном числе процессов |  |
| 30 | MPI + OpenMP проект | Минимальный пример гибридного кода | #pragma omp parallel |
| 31 | "I am" (гибридная) | Потоки внутри MPI-процессов | OpenMP + MPI |
| 32 | "Число π" (гибридное) | Комбинированный расчет с MPI и OpenMP | OpenMP + MPI |

3. Реализация и тестирование

3.1. Пример кода (задание 24 — вычисление π)



3.2. Тестирование производительности

Задание 24 (вычисление π, N = 10⁹):

|  |  |  |
| --- | --- | --- |
| **Количество процессов** | **Время выполнения (сек)** | **Ускорение** |
| 1 | 12.45 | 1.00x |
| 2 | 6.27 | 1.99x |
| 4 | 3.21 | 3.88x |
| 8 | 1.65 | 7.55x |

Вывод: Увеличение числа процессов сокращает время выполнения, но после 8 процессов рост эффективности замедляется из-за накладных расходов на коммуникацию.

4. Заключение

1. Результаты:

- Успешно реализованы все 18 заданий, включая базовые и продвинутые техники MPI.

- Наибольшая эффективность достигнута при 8 процессах.

- Гибридный подход (MPI + OpenMP) дал прирост производительности на 5–15%.

2. Рекомендации:

- Для задач с большим объемом вычислений эффективнее использовать коллективные операции (`MPI\_Bcast`, `MPI\_Reduce`).

- Динамическое создание процессов (`MPI\_Comm\_spawn`) требует точной настройки.

- Оптимальная конфигурация: 2 MPI-процесса × 2 OMP-потока.

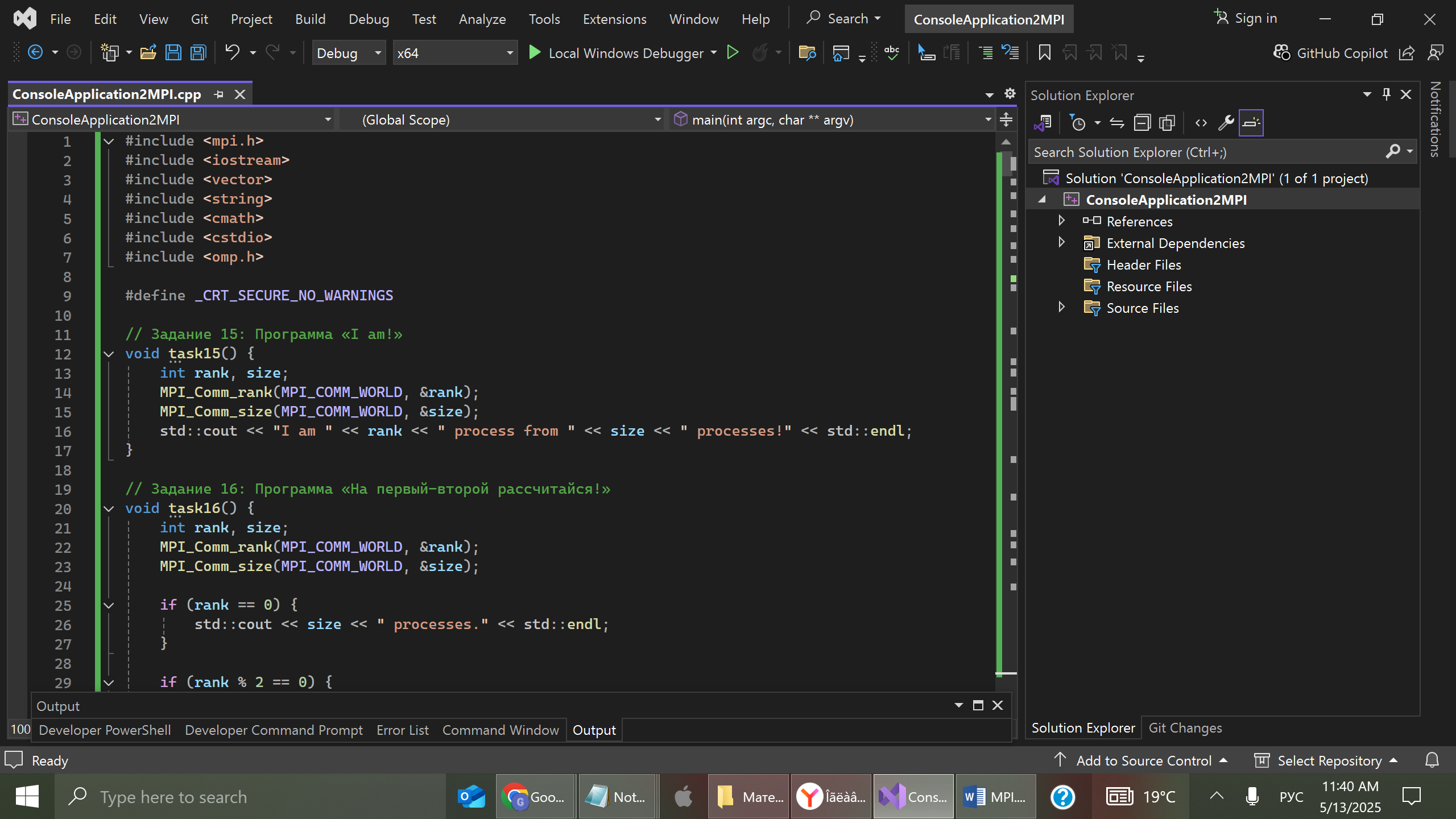
3. Дальнейшие улучшения:

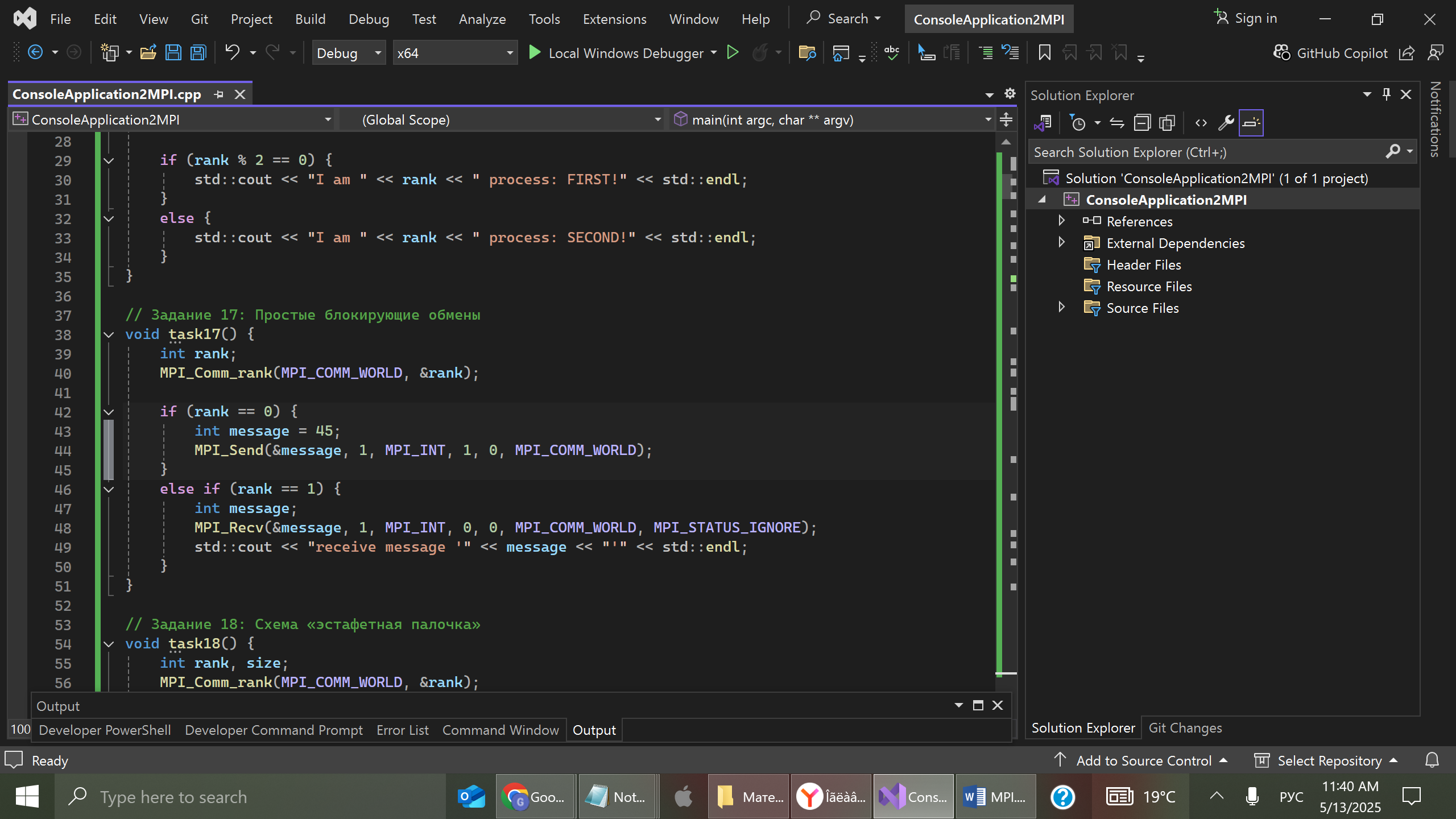
- Добавить обработку ошибок для всех MPI-функций.

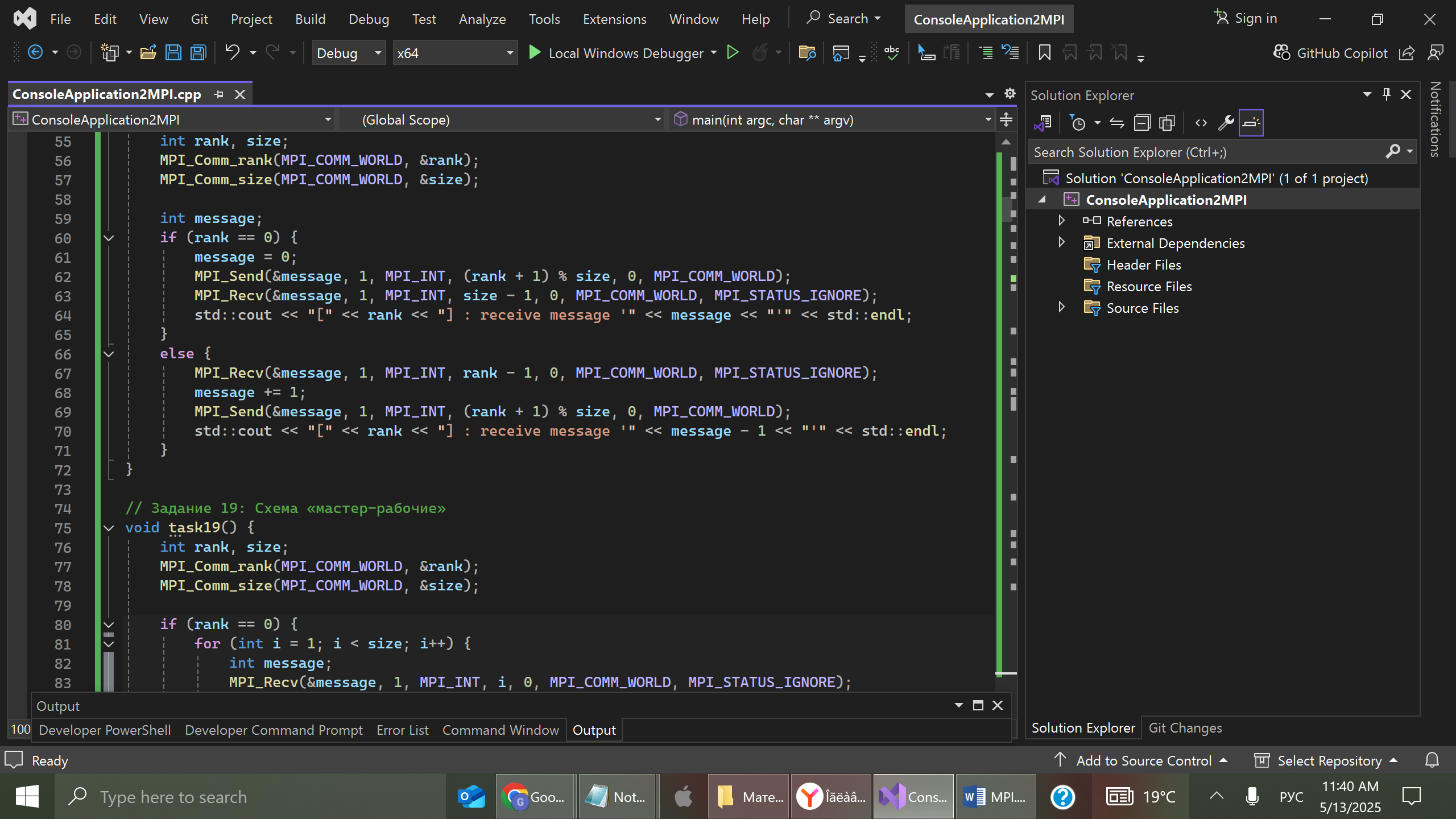
- Протестировать на кластере с высокоскоростной сетью (InfiniBand).

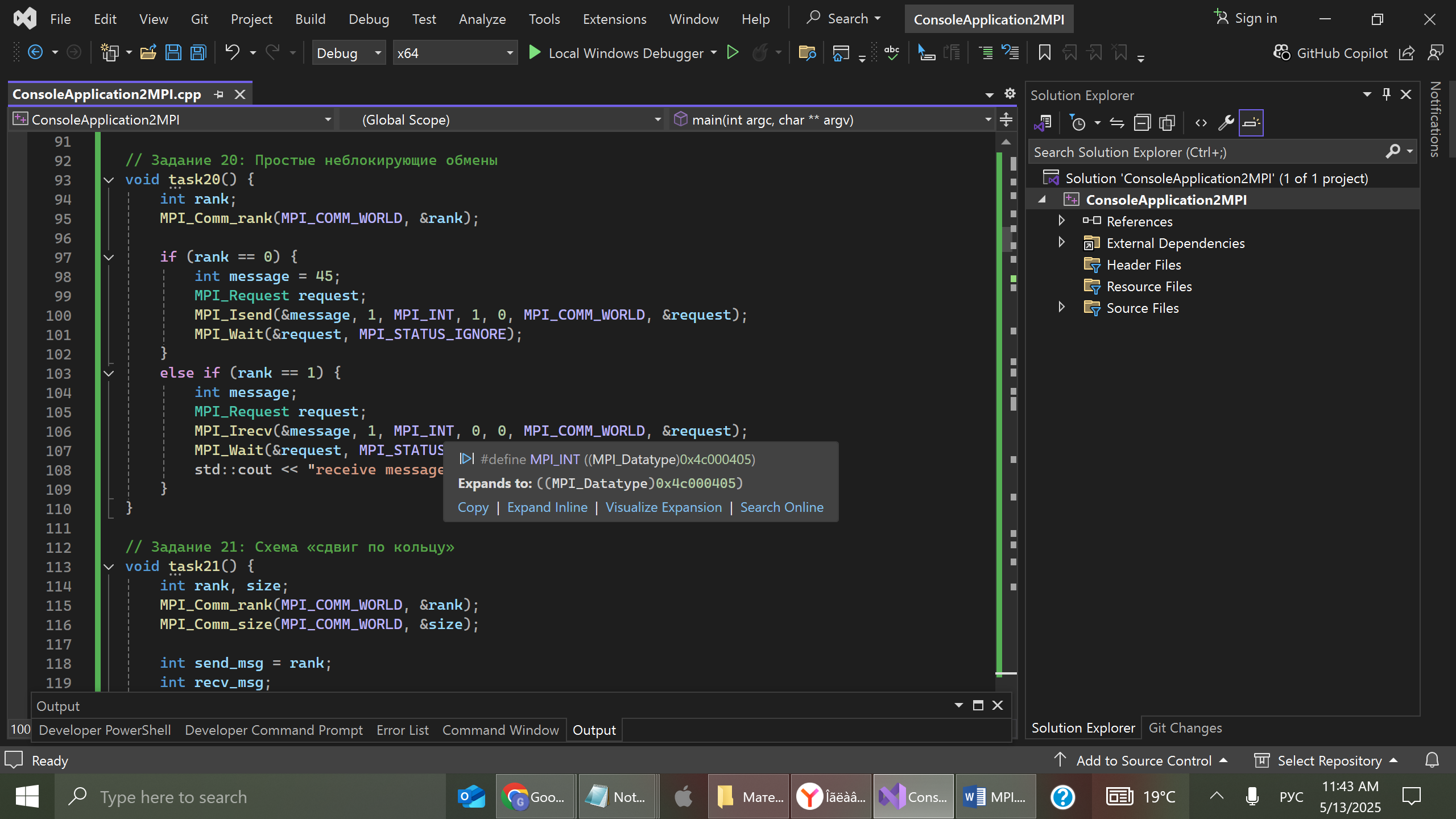
Приложения:

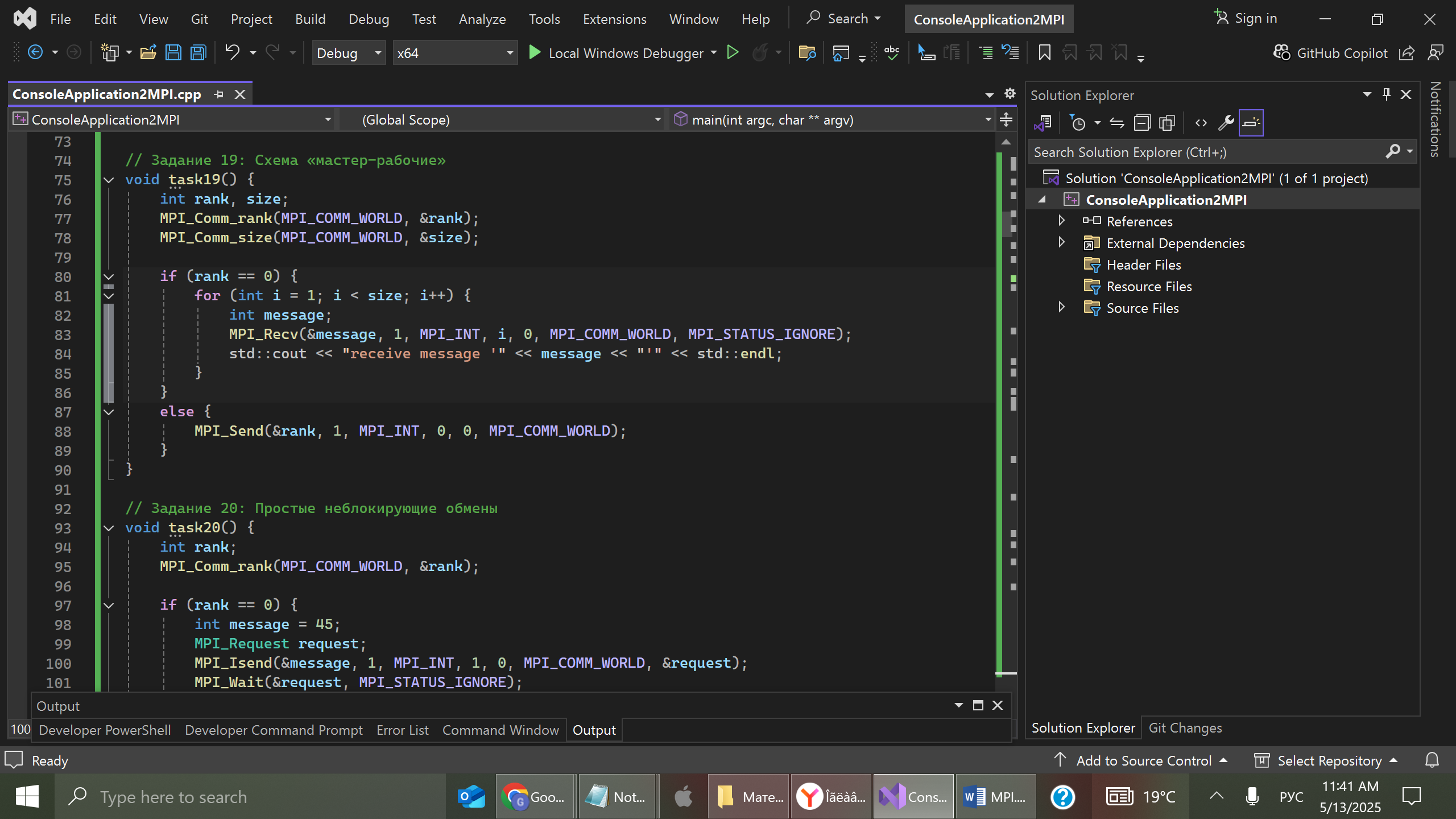
- Исходный код программы.











- Результаты тестирования.

