**Использование MPI для работы с текстовыми данными**

**Теоретический блок**

**Распределенная обработка текстовых данных:**

1. В реальных приложениях параллельные вычисления используются не только для численных задач, но и для обработки больших объемов текстовых данных, например:
   * Анализ логов.
   * Обработка файлов в поисковых системах.
   * Сортировка данных.

**Основные понятия:**

* **Разделение данных:** Большой текст делится на части, каждая из которых передается отдельному процессу.
* **Агрегация результатов:** После выполнения задачи процессы возвращают результаты в главный процесс.

**Задание**

**Задача:** Написать программу, которая подсчитывает количество уникальных слов в текстовом файле с использованием MPI.

**Описание:**

1. Главный процесс считывает текст из файла и делит его на равные части.
2. Каждому процессу передается его часть текста.
3. Каждый процесс подсчитывает уникальные слова в своей части.
4. Главный процесс собирает результаты и выводит общее количество уникальных слов.

**Требования к выполнению:**

1. Учесть обработку границ текстового файла, чтобы слова не разрывались между частями

**Ход выполнения:**

1. **Разделение данных:**
   * Прочитайте текст из файла в строку.
   * Разделите строку на подстроки, учитывая разрывы слов.
2. **Передача данных.**
3. **Обработка данных на процессах:**
   * На каждом процессе разбейте строку на слова.
4. **Сбор результатов:**
   * Соберите количество уникальных слов от каждого процесса.
5. **Вывод результатов:**
   * Главный процесс суммирует уникальные слова и выводит общий результат и время выполнения последовательного и параллельного алгоритма.

Программа разделит текст на части, подсчитает количество уникальных слов в каждой части, и главный процесс (ранг 0) выведет общий результат. Пример вывода:

*String from text.txt: MPI is a standard for parallel programming. Parallel programming is fun!*

*Total unique words: 8*

*Parallel time: … seconds*

*Sequence time: … seconds.*