

## Задания MPI

1. Написать программу, печатающую номер процесса и общее количество процессов в коммуникаторе MPI\_COMM\_WORLD.
2. Написать программу, запустить ее на 2х процессах. На нулевом процессе задать массив  $a$  из 10 элементов, значения сгенерировать случайным образом. Переслать этот массив целиком первому процессу с помощью функции MPI\_Send. Распечатать на первом процессе принятый массив.
3. Используя блокирующую операцию передачи сообщений (MPI\_Send и MPI\_Recv), выполнить пересылку данных одномерного массива, из процесса с номером 1 на остальные процессы группы. На процессах получателях до выставления функции приема (MPI\_Recv) определить количество принятых данных (MPI\_Probe). Выделить память под размер приемного буфера, после чего вызвать функцию MPI\_Recv. Полученное сообщение выдать на экран.
4. Пусть векторы  $x$ ,  $y$  заданы на нулевом процессе. Написать программу, в которой векторы равными блоками с нулевого процесса пересылаются остальным процессам. Пересылка данных осуществляется функцией MPI\_Send. Все процессы по формуле вычисляют свои элементы искомых массивов. Каждый процесс отправляет на нулевой процесс посчитанные элементы массивов. В программе реализовать следующие операции вычисления векторов:
  - a.  $z[i] = x[i] + y[i];$
  - b.  $z[i] = x[i] * y[i].$
5. Пусть на нулевом процессе задана матрица  $A$  и вектор  $x$ . Написать программу, в которой матрица равными блоками с нулевого процесса пересылается остальным процессам. Пересылка данных осуществляется функцией MPI\_Send. Все процессы по формуле вычисляют свои элементы искомых массивов. Каждый процесс отправляет на нулевой процесс посчитанные элементы массивов. В программе реализовать операцию  $x = \text{diag}(A)$  - выделения главной диагонали.
6. Разработать программу, в которой два процесса многократно обмениваются сообщениями длиной  $n$  байт. Выполнить эксперименты и оценить зависимость времени выполнения операции от длины сообщения. Рассмотреть два варианта обмена сообщениями
  - a. PingPong - двунаправленная передача. Выполняется передача данных от одного процесса другому с последующим возвратом переданных данных в исходный процесс. Время выполнение замеряется после получения обратного ответа на исходном процессе.
  - b. PingPing - однонаправленная передача. Два процесса одновременно выполняют передачу данных друг другу. Время выполнение замеряется после получения сообщения каждым процессом.
7. Написать программу вычисления нормы  $\|x\|_1 = \sum_{i=0}^{n-1} |x_i|$  вектора  $x$ . Для распределения элементов вектора  $x$  по процессам использовать функцию MPI\_Scatter. Для получения итоговой суммы на нулевом процессе использовать функцию MPI\_Reduce с операцией MPI\_Sum.

8. Написать программу вычисления скалярного произведения двух векторов. Для распределения элементов векторов  $x$ ,  $y$  по процессам использовать функцию `MPI_Scatter`. Для получения итоговой суммы на нулевом процессе использовать функцию `MPI_Reduce` с операцией `MPI_Sum`.
9. В массиве вещественных чисел вычислить минимальное значение и глобальный индекс минимального элемента. Использовать функции `MPI_Scatter` и `MPI_Reduce` с операцией `MPI_MINLOC`.
10. Написать программу умножения матрицы на вектор  $z = Ax$ . Распределить матрицу  $A$  равномерно вдоль строк, вектор  $x$  передать всем процессам. После выполнения операции умножения матрицы на вектор на каждом процессе необходимо выполнить сбор полученных локальных частей результирующего вектора на один из процессов. Использовать функцию `MPI_Bcast` для рассылки вектора  $x$ , для распределения элементов матрицы использовать функцию `MPI_Scatter`, для сбора локальных частей результирующего вектора в глобальный вектор использовать функцию `MPI_Gather`.
11. На процессе с номером 0 объявить и заполнить значениями матрицу  $a[8][8]$ . Создать новый тип данных для рассылки нечетных строк матрицы  $a$  в матрицу  $b[4][8]$ , и для рассылки четных строк матрицы в матрицу  $c[4][8]$ , используя функцию `MPI_Type_vector`.
12. Создать тип данных с помощью функции `MPI_Type_create_struct` и выполнить рассылку строк матрицы  $a[8][8]$  на 4 процесса в матрицу  $d[2][8]$  по следующей схеме:
- a. на нулевой процесс строки 0 и 4,
  - b. на первый процесс строки 1 и 5,
  - c. на второй процесс строки 2 и 6,
  - d. на третий процесс строки 3 и 7.
- Пример работы с функцией `MPI_Type_create_struct` см. здесь  
[https://docs.google.com/document/d/1vjyKSQEAenjw85\\_jD9SIjJDyUXBUfX1IkUVHGnLOA0Q/edit?usp=sharing](https://docs.google.com/document/d/1vjyKSQEAenjw85_jD9SIjJDyUXBUfX1IkUVHGnLOA0Q/edit?usp=sharing)
13. На одном из процессов объявить и заполнить значениями верхнюю треугольную матрицу. Создать новый тип данных для передачи ненулевых элементов матрицы, используя конструктор `MPI_Type_indexed`. Выполнить передачу элементов на все процессы с помощью функции `MPI_Bcast`.
14. Пусть на нулевом процессе определены и заданы массив из десяти целых чисел и массив из восьми вещественных чисел. Упаковать два массива. Передать упакованные данные на все процессы. Распаковать на всех процессах принятые данные. Результат выдать на экран.