

# Курс "Вычисления на многопроцессорных компьютерах"

## Лабораторный практикум

### Лабораторная работа №2:

## Освоение процедур приема-передачи данных между параллельно исполняющимися процессами в MPI

**Цель работы** - освоить принципы работы процедур приема-передачи сообщений в MPI, реализующими двухточечные обмены (с блокировкой и без блокировки) и коллективные обмены данными.

### Варианты заданий

- 1. Передача сообщения нулевым процессом остальным с помощью двухточечных коммуникационных операций**  
Написать программу, в которой нулевой процесс инициализирует некоторую переменную (например, строку "Hello, World!") и выполняет блокирующую и неблокирующую передачу этой переменной всем остальным исполняющим программу процессам с использованием процедур MPI\_Send()/MPI\_Recv() и MPI\_Isend()/MPI\_Irecv(), соответственно. Для неблокирующего обмена напечатать результаты выполнения передачи данных (значение передаваемой переменной в принимающих процессах) на моменты до и после применения процедуры MPI\_Wait(). Провести запуск программ с разным задаваемым числом процессов и проанализировать результаты выполнения.
- 2. Двухточечный обмен сообщениями с использованием различных вариантов выполнения передачи**  
Написать программу, в которой нулевой процесс инициализирует и организует передачу сообщения (например, вещественной переменной) процессу с номером один. Поэкспериментировать с различными способами реализации операций обмена (блокирующей и неблокирующей), в том числе задавая разные типы процедур приема/передачи в рамках одной и той же операции двухточечного обмена. Проанализировать результаты выполнения программы, выводя на печать значения переменной в передающем и принимающем процессах.
- 3. Блокирующий двусторонний обмен сообщениями между двумя процессами**  
Написать программу, в которой нулевой процесс организует блокирующую (с помощью процедур MPI\_Send()/MPI\_Recv()) передачу сообщения (например, вещественной переменной с именем b) процессу с номером один и ждет от него ответного сообщения (например, вещественную переменную с именем a). Напечатать номера процессов и значения переменных a и b в передающем, принимающем сообщения процессах, а также в процессах, не участвующих в обмене. При запусках программы назначить число процессов, равное двум и более. Начальные значения переменных a и b сделать отличающимися от передаваемых. Проанализировать результаты выполнения программы.
- 4. Неблокирующий обмен сообщениями между всеми процессами в соответствии с топологией кольца**  
Написать программу, в которой все исполняющие ее процессы при помощи неблокирующих коммуникационных операций (процедуры MPI\_Isend()/MPI\_Irecv()) однократно обмениваются сообщениями с ближайшими соседями (по номерам процессов) в соответствии с топологией кольца. Напечатать значения передаваемых переменных (например, номера процесса-отправителя) после получения сообщений. Реализовать проверку состояния приема с помощью процедур MPI\_Wait() и MPI\_Test().
- 5. Определение структуры приходящих сообщений при организации двухточечных обменов**  
Написать программу, в которой нулевой процесс ожидает сообщения разного типа (например, целочисленная и вещественная переменные), но с одинаковым идентификатором от любого из процессов с номерами 1 и 2. Принять сообщения, предварительно определив, от какого из процессов пришло каждое из них (использовать процедуру MPI\_Probe()). Напечатать принятые нулевым процессом сообщения и номера процессов-отправителей.
- 6. Отложенные множественные двухточечные обмены сообщениями**  
Написать программу, реализующую отложенный обмен несколькими заранее подготовленными

сообщениями между двумя процессами из группы (использовать процедуры `MPI_Send_init()` и `MPI_Startall()`).

7. *Реализация вычислительной схемы "master-slave" с использованием процедуры `MPI_Waitsome()`*  
Написать программу, в которой все, за исключением одного, процессы из группы ("slave") непрерывно "общаются" с одним выделенным главным процессом ("master") в рамках вычислительной схемы "master-slave". При этом все процессы, кроме главного ("master"), на каждой итерации глобального для всех процессов условно бесконечного цикла передачи назначают (путем вызова определенной пользователем функции обработки данных - "slave") свою локальную часть некоторого массива данных, после чего посылают ее главному процессу. Главный процесс исходно инициализирует неблокирующие приемы от всех остальных процессов, после чего дожидается прихода хотя бы одного, любого сообщения от любого процесса (использовать процедуру `MPI_Waitsome()`). Для пришедших сообщений главный процесс вызывает некоторую процедуру обработки данных "master", после чего снова выставляет неблокирующие приемы.
8. *Рассылка данных при помощи коллективных коммуникационных операций*  
Написать программу, в которой нулевой процесс инициализирует двумерный массив (матрицу) вещественных чисел и рассылает ее, разбивая последовательно на столбцы, другим процессам приложения (использовать процедуру `MPI_Scatter()`). После рассылки в процессах производится однотипное изменение элементов массива (например, умножение на 2). Напечатать исходный и модифицированный массив данных.
9. *Сбор распределенных данных при помощи коллективных коммуникационных операций*  
Написать программу, в которой организуется сбор по 10 вещественных целых чисел от каждого процесса группы в нулевом процессе (использовать процедуру `MPI_Gather()`). Напечатать принятые нулевым процессом наборы данных и номера процессов-отправителей.
10. *Сбор распределенных данных при помощи коллективных коммуникационных операций (векторной процедуры сбора)*  
Написать программу, в которой организуется сбор 10 целых чисел от каждого процесса группы в нулевой процесс. При этом каждый набор из 10 чисел размещается с некоторым смещением ("страйдом") относительно конца предыдущего набора (использовать функцию `MPI_Gatherv()`).
11. *Рассылка распределенных данных при помощи коллективных коммуникационных операций (векторной процедуры рассылки)*  
Написать программу, в которой нулевой процесс инициализирует массив из целых чисел, после чего организует рассылку наборами по 10 подряд идущих элементов массива последовательно остальным процессам группы, начиная с первого. При этом каждый набор из 10 элементов размещается в посылающем буфере с некоторым смещением ("страйдом") относительно конца предыдущего набора (использовать процедуру `MPI_Scatterv()`).
12. *Сбор распределенных данных во всех процессах при помощи коллективных коммуникационных операций*  
Написать программу, в которой организуется сбор по 10 целых чисел от каждого процесса группы в каждом процессе, так что в итоге у всех процессов оказываются одинаковые числовые наборы (использовать процедуру `MPI_Allgather()`).
13. *Широковещательная рассылка данных*  
Написать программу, реализующую широковещательную рассылку данных (например, массива вещественных чисел) от нулевого процесса всем процессам группы (использовать процедуру `MPI_Bcast()`).
14. *Глобальная вычислительная операция над распределенными данными (операция редукции)*  
Написать программу, реализующую стандартную глобальную вычислительную операцию (например, сложения) над распределенными по процессам данными (использовать процедуру `MPI_Reduce()`).
15. *Глобальная вычислительная функциональная операция над распределенными данными (операция редукции)*  
Написать программу, в которой некоторая определенная пользователем функция используется в качестве глобальной вычислительной операции (редукции).
16. *Глобальная вычислительная операция над распределенными данными: моделирование с помощью других процедур стандарта*  
Программно смоделировать процедуру `MPI_Allreduce()` при помощи процедур `MPI_Reduce()` и `MPI_Bcast()`.
17. *Барьерная синхронизация процессов*  
Реализовать барьерную синхронизацию процессов при помощи двухточечных обменов и с использованием процедуры `MPI_Barrier()`.

## Структура отчета по лабораторной работе

По результатам выполнения лабораторной работы необходимо подготовить отчет, который должен включать следующие разделы:

- название лабораторной работы и текст задания к ней
  - исходные коды разработанных программ с содержательными комментариями к использованным процедурам MPI
  - результаты работы программ в соответствии с текстом заданий
  - отдельное пояснение необходимости выполнения операций обмена в MPI-программах
  - общие выводы по лабораторной работе
- 

### Учебные материалы:

[Лекции по технологии MPI](#)

### Книги и учебники:

[Антонов А.С. Параллельное программирование с использованием технологии MPI](#)

[Баканов В.М., Осипов Д.В. Введение в практику разработки параллельных программ в стандарте MPI](#)

[Шпаковский Г.И., Серикова Н.В. Программирование для многопроцессорных систем в стандарте MPI](#)

[Гришагин В.А., Свистунов А.Н. Параллельное программирование на основе MPI](#)

[MPI Tutorial \(примеры программ на Fortran\)](#)

[MPI: The Complete Reference](#) (руководство по MPI с примерами)

[Бартеньев О.В. Современный Фортран](#)

[Немнюгин М.А., Стесик О.Л. Современный Фортран. Самоучитель](#)

---

### Примеры программ:

- [MPI-программа вычисления определенного интеграла](#)
  - [MPI-программа с организацией двухточечных обменов с блокировкой](#)
  - [MPI-программа с организацией двухточечных обменов без блокировки](#)
  - [MPI-программа, использующая процедуру рассылки данных MPI\\_Scatter](#)
  - [Примеры программ из учебника Антонова А.С. "Технологии параллельного программирования MPI и OpenMP"](#)
  - [Другие примеры исходных кодов программ](#)
-