

Процессы и их ранги

1. В каждом из процессов, входящих в коммунитор MPI_COMM_WORLD, прочесть одно вещественное число и вывести его противоположное значение.
2. В каждом из процессов, входящих в коммунитор MPI_COMM_WORLD, прочесть одно целое число A и вывести его удвоенное значение. Кроме того, для главного процесса (процесса ранга 0) вывести количество процессов, входящих в коммунитор MPI_COMM_WORLD.
3. MPI1Proc3. В главном процессе прочесть вещественное число X и вывести его противоположное значение, в каждом из остальных процессов (подчиненных процессов, ранг которых больше 0) вывести его ранг.
4. MPI1Proc4. В процессах четного ранга (включая главный) ввести целое число и вывести его удвоенное значение. В процессах нечетного ранга не выполнять никаких действий.
5. В процессах четного ранга (включая главный) ввести целое число, в процессах нечетного ранга ввести вещественное число. В каждом процессе вывести удвоенное значение введенного числа.
6. В подчиненных процессах четного ранга ввести целое число, в процессах нечетного ранга ввести вещественное число. В каждом подчиненном процессе вывести удвоенное значение введенного числа. В главном процессе не выполнять никаких действий.
7. В каждом процессе четного ранга (включая главный) дано целое число N (> 0) и набор из N вещественных чисел. Вывести в каждом из этих процессов сумму чисел из данного набора. В процессах нечетного ранга не выполнять никаких действий.
8. В каждом процессе дано целое число N (> 0) и набор из N вещественных чисел. В процессах четного ранга (включая главный) вывести сумму чисел из данного набора, в процессах нечетного ранга вывести среднее арифметическое чисел из данного набора.
9. В каждом процессе дано целое число N (> 0) и набор из N вещественных чисел. В подчиненных процессах четного ранга вывести сумму чисел из данного набора, в процессах нечетного ранга вывести среднее арифметическое чисел из данного набора, в главном процессе вывести произведение чисел из данного набора.
10. В каждом процессе дано целое число N (> 0) и набор из N чисел, причем в подчиненных процессах нечетного ранга (1, 3, ...) набор содержит вещественные числа, в подчиненных процессах четного ранга (2, 4, ...) — целые числа, а тип элементов в главном процессе зависит от общего количества процессов: если количество процессов нечетное, то набор содержит целые числа, а если четное, то вещественные. В процессах четного ранга (включая главный) вывести минимальный элемент из данного набора, в процессах нечетного ранга вывести максимальный элемент.
11. В подчиненных процессах четного ранга ввести целое число, в процессах нечетного ранга ввести вещественное число. В каждом подчиненном процессе вывести квадрат и куб введенного числа. В главном процессе не выполнять никаких действий.
12. В каждом процессе четного ранга (включая главный) дано целое число N (> 0) и набор из N вещественных чисел. Вывести в каждом из этих процессов произведение чисел из данного набора. В процессах нечетного ранга не выполнять никаких действий.
13. В каждом процессе дано целое число N (> 0) и набор из N вещественных чисел. В процессах четного ранга (включая главный) вывести последовательную разность чисел из данного набора, в процессах нечетного ранга вывести среднее арифметическое геометрическое чисел из данного набора.
14. В каждом процессе дано целое число N (> 0) и набор из N вещественных чисел. В подчиненных процессах четного ранга вывести произведение чисел из данного набора, в процессах нечетного ранга вывести сумму квадратов чисел из данного набора, в главном процессе вывести произведение чисел из данного набора.
15. В каждом процессе дано целое число N (> 0) и набор из N чисел, причем в подчиненных процессах нечетного ранга (1, 3, ...) набор содержит вещественные числа, в подчиненных процессах четного ранга (2, 4, ...) — целые числа, а тип элементов в главном процессе зависит от общего количества процессов: если количество процессов нечетное, то набор содержит целые числа, а если четное, то вещественные. В процессах четного ранга (включая главный) вывести минимальный по модулю элемент из данного набора, в процессах нечетного ранга вывести максимальный по модулю элемент.

Блокирующая пересылка данных

1. В каждом подчиненном процессе дано целое число. Переслать эти числа в главный процесс, используя функции MPI_Send и MPI_Recv (стандартные блокирующие функции для передачи и приема сообщения), и вывести их в главном процессе. Полученные числа выводить в порядке возрастания рангов переславших их процессов.
2. В каждом подчиненном процессе дано вещественное число. Переслать эти числа в главный процесс, используя функции MPI_Bsend (посылка сообщения с буферизацией) и MPI_Recv, и вывести их в главном процессе. Полученные числа выводить в порядке убывания рангов переславших их процессов. Для задания буфера использовать функцию MPI_Buffer_attach.
3. В каждом подчиненном процессе даны четыре целых числа. Переслать эти числа в главный процесс, используя по одному вызову функции MPI_Send для каждого передающего процесса, и вывести их в главном процессе. Полученные числа выводить в порядке возрастания рангов переславших их процессов.
4. В каждом подчиненном процессе дано целое число N ($0 < N < 5$) и набор из N целых чисел. Переслать данные наборы в главный процесс, используя по одному вызову функции MPI_Bsend для каждого передающего процесса, и вывести наборы в главном процессе в порядке возрастания рангов переславших их процессов. Для определения размера пересланного набора использовать функцию MPI_Get_count.
5. В главном процессе дан набор вещественных чисел; количество чисел равно количеству подчиненных процессов. С помощью функции MPI_Send переслать по одному числу в каждый из подчиненных процессов (первое число в процесс 1, второе — в процесс 2, и т. д.) и вывести в подчиненных процессах полученные числа.
6. В главном процессе дан набор вещественных чисел; количество чисел равно количеству подчиненных процессов. С помощью функции MPI_Bsend переслать по одному числу в каждый из подчиненных процессов, перебирая процессы в обратном порядке (первое число в последний процесс, второе — в предпоследний процесс, и т. д.), и вывести в подчиненных процессах полученные числа.
7. В главном процессе дано целое число N и набор из N чисел; $K - 1 \leq N < 10$, где K — количество процессов. С помощью функции MPI_Send переслать по одному числу из данного набора в процессы 1, 2, ..., $K - 2$, а оставшиеся числа — в процесс $K - 1$, и вывести полученные числа. В процессе $K - 1$ для определения количества полученных чисел использовать функцию MPI_Get_count.
8. В каждом подчиненном процессе дано целое число, причем только для одного процесса это число отлично от нуля. Переслать ненулевое число в главный процесс и вывести в главном процессе полученное число и ранг процесса, переславшего это число. Для приема сообщения в главном процессе использовать функцию MPI_Recv с параметром MPI_ANY_SOURCE.
9. В каждом подчиненном процессе дано целое число N , причем для одного процесса это число больше нуля, а для остальных равно нулю. В процессе с ненулевым N дан также набор из N чисел. Переслать данный набор чисел в главный процесс и вывести в главном процессе полученные числа и ранг процесса, переславшего этот набор. При приеме сообщения использовать параметр MPI_ANY_SOURCE.
10. В каждом подчиненном процессе дано целое число N , в главном процессе дано целое число K (> 0), равное количеству тех подчиненных процессов, в которых даны положительные числа N . Переслать все положительные числа N в главный процесс и вывести в нем сумму полученных чисел. Для приема сообщений в главном процессе использовать функцию MPI_Recv с параметром MPI_ANY_SOURCE.
11. В каждом процессе дано вещественное число. Переслать число из главного процесса во все подчиненные процессы, а все числа из подчиненных процессов — в главный, и вывести в каждом процессе полученные числа (в главном процессе числа выводить в порядке возрастания рангов переславших их процессов). Для отправки сообщений использовать функцию MPI_Ssend. **Замечание.** Функция MPI_Ssend обеспечивает синхронный режим пересылки данных, при котором операция отправки сообщения будет завершена только после начала приема этого сообщения процессом-получателем. В случае пересылки данных в синхронном режиме возникает опасность взаимных

блокировок (deadlocks) из-за неправильного порядка вызова функций отправки и получения сообщений.

12. В каждом процессе дано целое число. С помощью функций MPI_Ssend и MPI_Recv осуществить для всех процессов циклический сдвиг данных с шагом 1, переслав число из процесса 0 в процесс 1, из процесса 1 в процесс 2, ..., из последнего процесса в процесс 0. В каждом процессе вывести полученное число.

13. В каждом процессе дано целое число. С помощью функций MPI_Ssend и MPI_Recv осуществить для всех процессов циклический сдвиг данных с шагом -1, переслав число из процесса 1 в процесс 0, из процесса 2 в процесс 1, ..., из процесса 0 в последний процесс. В каждом процессе вывести полученное число.

14. В каждом процессе даны два целых числа. С помощью функций MPI_Ssend и MPI_Recv переслать первое число в предыдущий процесс, а второе — в последующий процесс (для процесса 0 считать предыдущим последний процесс, а для последнего процесса считать последующим процесс 0). В каждом процессе вывести числа, полученные от предыдущего и последующего процесса (в указанном порядке).

15. В каждом процессе даны два числа: вещественное A и целое N, причем набор чисел N содержит все значения от 0 до K - 1, где K — количество процессов. Используя функции MPI_Send и MPI_Recv (с параметром MPI_ANY_SOURCE), выполнить в каждом процессе пересылку числа A в процесс N и вывести полученное число, а также ранг процесса, из которого число было получено.