**Программирование на основе MPI. Парные операции приема-передачи сообщений. Асинхронные операции приема-передачи сообщений. Отложенные запросы.**

**Передача сообщений**

Для *передачи сообщения*процесс-отправитель должен выполнить функцию:

int MPI\_Send(void \*buf, int count, MPI\_Datatype type, int dest,

int tag, MPI\_Comm comm),

где

* buf - адрес буфера памяти, в котором располагаются данные отправляемого сообщения,
* count - количество элементов данных в сообщении,
* type - тип элементов данных пересылаемого сообщения,
* dest - ранг процесса, которому отправляется сообщение,
* tag - значение-тег, используемое для идентификации сообщений,
* *comm* - *коммуникатор*, в рамках которого выполняется передача данных.

Для указания типа пересылаемых данных в *MPI* имеется ряд базовых типов, полный список которых приведен в [табл. 6.1](https://intuit.ru/studies/courses/542/398/lecture/9177?page=2#table.6.1).

|  |  |
| --- | --- |
| Таблица 6.1. Базовые (пpедопpеделенные) типы данных *MPI* для *алгоритмического языка* C | |
| **MPI\_Datatype** | **C*****Datatype*** |
| MPI\_BYTE |  |
| MPI\_CHAR | signed char |
| MPI\_DOUBLE | double |
| MPI\_FLOAT | float |
| MPI\_INT | int |
| MPI\_LONG | long |
| MPI\_LONG\_DOUBLE | long double |
| MPI\_PACKED |  |
| MPI\_SHORT | short |
| MPI\_UNSIGNED\_CHAR | *unsigned* char |
| MPI\_UNSIGNED | *unsigned* int |
| MPI\_UNSIGNED\_LONG | *unsigned* long |
| MPI\_UNSIGNED\_SHORT | *unsigned* short |

##### Прием сообщений

Для *приема сообщения*процесс-получатель должен выполнить функцию:

int MPI\_Recv(void \*buf, int count, MPI\_Datatype type, int source,

int tag, MPI\_Comm comm, MPI\_Status \*status),

где

* buf, count, type - буфер памяти для приема сообщения, назначение каждого отдельного параметра соответствует описанию в MPI\_Send,
* source - ранг процесса, от которого должен быть выполнен прием сообщения,
* tag - тег сообщения, которое должно быть принято для процесса,
* *comm* - *коммуникатор*, в рамках которого выполняется передача данных,
* status - указатель на структуру данных с информацией о результате выполнения операции приема данных.

#### **Организация неблокирующих обменов данными между процессорами (асинхронные)**

int MPI\_Isend(void \*buf, int count, MPI\_Datatype type, int dest,

int tag, MPI\_Comm comm, MPI\_Request \*request)

int MPI\_Irecv(void \*buf, int count, MPI\_Datatype type, int source,

int tag, MPI\_Comm comm, MPI\_Request \*request)

MPI\_ISEND(BUF, COUNT, DATATYPE, DEST, MSGTAG, COMM, REQUEST, IERR)

<type> BUF(\*)

INTEGER COUNT, DATATYPE, DEST, MSGTAG, COMM, REQUEST, IERR

Неблокирующая посылка из буфера BUF COUNT элементов сообщения типа *DATATYPE* с идентификатором MSGTAG процессу DEST *коммуникатора* сомм.

MPI\_IRECV(BUF, COUNT, DATATYPE, SOURCE, MSGTAG, COMM, REQUEST, IERR)

<type> BUF(\*)

INTEGER COUNT, DATATYPE, SOURCE, MSGTAG, COMM, REQUEST, IERR

Неблокирующий прием в буфер BUF не более COUNT элементов сообщения типа *DATATYPE* с идентификатором MSGTAG от процесса с номером SOURCE В *коммуникаторе* сомм с заполнением массива STATUS.

##### Отложенные запросы

Часто в программе приходится многократно выполнять обмены с одинаковыми параметрами (например, в цикле). В этом случае можно один раз инициализировать операцию обмена и потом многократно ее запускать, не тратя на каждой итерации дополнительного времени на инициализацию и заведение соответствующих внутренних структур данных. Кроме того, таким образом, несколько запросов на прием и/или передачу могут объединяться вместе для того, чтобы далее их можно было бы запустить одной командой

MPI\_SEND\_INIT(BUF, COUNT, DATATYPE, DEST, MSGTAG, COMM, REQUEST,

IERR)

<type> BUF(\*)

INTEGER COUNT, DATATYPE, DEST, MSGTAG, COMM, REQUEST, IERR

Формирование отложенного запроса на посылку сообщения. Сама операция пересылки при этом не начинается!

Аналогично трем модификациям процедур MPI\_SEND И MPI\_ISEND, предусмотрены три дополнительных варианта процедуры MPI\_SEND\_INIT:

* MPI\_BSEND\_INIT - формирование отложенного запроса на передачу сообщения с буферизацией;
* MPI\_SSEND\_INIT - формирование отложенного запроса на передачу сообщения с синхронизацией;
* MPI\_RSEND\_INIT - формирование отложенного запроса на передачу сообщения по готовности.

MPI\_RECV\_INIT(BUF, COUNT, DATATYPE, SOURCE, MSGTAG, COMM,

REQUEST, IERR)

<type> BUF(\*)

INTEGER COUNT, DATATYPE, SOURCE, MSGTAG, COMM, REQUEST, IERR

Формирование отложенного запроса на прием сообщения. Сама операция приема при этом не начинается!

MPI\_START(REQUEST, IERR) INTEGER REQUEST, IERR

Инициализация отложенного запроса на выполнение операции обмена, соответствующей значению параметра REQUEST. Операция запускается как неблокирующая.

MPI\_STARTALL(COUNT, REQUESTS, IERR) INTEGER COUNT, REQUESTS, IERR

Инициализация COUNT отложенных запросов на выполнение операций обмена, соответствующих значениям первых COUNT элементов массива REQUESTS. Операции запускаются как неблокирующие.

В отличие от неблокирующих операций, по завершении выполнения операции, запущенной при помощи отложенного запроса на взаимодействие, значение параметра REQUEST (REQUESTS) сохраняется и может использоваться в дальнейшем!

MPI\_REQUEST\_FREE(REQUEST, IERR) INTEGER REQUEST, IERR

Данная процедура удаляет структуры данных, связанные с параметром REQUEST. После ее выполнения параметр REQUEST устанавливается в значение MPI\_REQUEST\_NULL. Если операция, связанная с этим запросом, уже выполняется, то она будет завершена.