1. **Как расшифровать аббревиатуру MPI?**

Message-Passing-Interface

1. **Назовите известные реализации стандарта MPI.**

MPI-1.0, MPI-2.0, MPI-3.0, MPI-3.1

1. **Какой минимальный набор из шести функций среды MPI позволяет**

**создать параллельную программу с обменом данными между процессами?**

* инициализация и завершение приложения;
* определение ранга процесса и общего числа процессов;
* организация приема и передачи сообщения от одного процесса другому.

1. **Можно ли вызвать функции среды MPI вне параллельной области   
   программы? Если да, то какие это функции? Если нет, то почему нельзя?**

Вне параллельной области программы вызов функции MPI (MPI\_Init())запрещен.

Можно вызывать:

MPI\_GET\_version() - позволяет получить информацию о поддерживаемой реализацией версии стандарта MPI

MPI\_Get\_library\_version() () - позволяет получить информацию о поддерживаемой реализацией версии библиотеки

MPI\_Initialized() – может быть использована для проверки, исполняется ли данный программный код параллельно

MPI\_Finalised() – может быть использована для проверки, исполняется ли данный программный код параллельно

1. **Существует ли способ узнать версию стандарта MPI средствами параллельной программы?**

ДА. MPI\_GET\_version() - позволяет получить информацию о поддерживаемой реализацией версии стандарта MPI

1. **Существует ли возможность узнать версию используемой библиотеки MPI средствами параллельной программы?**

ДА. MPI\_Get\_library\_version() () - позволяет получить информацию о поддерживаемой реализацией версии библиотеки

1. **Дайте определение термину « коммуникатор».**

Коммуникатор – группа процессов, которой поставлен в соответствие дескриптор(специальное описание объекта)

1. **Верно ли утверждение: «все функции MPI возвращают значение целочисленного типа int »? Если оно верно, то дайте обоснование существованию такого правила. Если не верно, то какие исключения известны?**

Нет, не верно. Все функции среды MIP (за исключением MPI\_Wtime() и MPI\_Wtick(), это функции для работы с таймером) возвращают значение типа int. В случае успешного завершения любой функции возвращается значение MPI\_SUCCESS, при наличии ошибки возвращаемое значение зависит от используемой реализации стандарта MPI.

1. **Какие функции используются для попарного обмена сообщениями?**

MPI\_Send(), MPI\_Recv()

1. **Для чего в стандарте MPI предусмотрены групповые имена?**

Для принятия сообщений от произвольного источника (не конкретизировать ранг процесса-отправителя)

1. **Как измерить время работы вычислительной части программного кода средствами MPI?**

double t1, t2;

t1 = MPI\_Wtime();

//Вычислительная часть программы

t2 = MPI\_Wtime();

1. **Перечислите предусмотренные стандартом MPI способы попарного обмена сообщениями.**

* буферизированный способ MPI\_Bsend()
* синхронный MPI\_Ssend()
* способ передачи данных «по-готовности» MPI\_Rsend()

1. **В каких случаях уместно применять неблокирующие операции взаимодействия между процессами?**

Их применение уместно в системах, где взаимодействие между процессорами может быть выполнено автономно по отношению к выполнению вычислительных операций

1. **С помощью какой функции выполняется операция синхронизации процессов?**

MPI\_Barrier(**)**

1. **Опишите действие функций: broadcast, gather, scatter, allgather, alltoall.**

broadcast – широковещательная рассылка данных заключается в передачи сообщения от процесса-отправителя с рангом root всем процессам, входящим в коммуникатор comm. MPI\_Broadcast()

gather – сбор данных на одном процессе. Каждый процесс отправляет содержимое локального буфера одному процессу-получателю с рангом root. MPI\_Gather()

scatter – распределение данных от одного процесса. MPI\_Scatter()

allgather – сбор данных на всех процессах. Операция сбора данных на всех процессах похожа на MPI\_Gather(), только тут все процессы коммуникатора, не только root, получают данные. MPI\_Allgather()

alltoall – распределение данных от всех процессов всем процессам. Обобщает операцию allgather, каждый процесс пересылает различные данные другим процессам. MPI\_Alltoall()

1. **В каких случаях применяются «векторные» варианты коллективных операций?**

Векторные варианты применяют в качестве одного из аргументов не размер сообщения, а массив таких размеров

1. **Опишите, как в среде MPI организовать операцию редукции.**

Широко применяемая в параллельных приложениях операция редукции MPI\_Reduce() вычисляет определенную функцию от данных, полученных одним из процессов от всех процессов коммуникатора:

int MPI\_Reduce(const void \* sendbuf, void \* recvbuf, int count,

MPI\_Datatype datatype, MPI\_Op op, int root, MPI\_Comm comm)

Используются следующие аргументы:

sendbuf - адрес буфера памяти с исходными данными;

recvbuf - адрес буфера памяти, принимающего данные;

count - количество элементов в массиве, адресуемом sendbuf;

datatype - тип пересылаемых данных;

ор - функция, применяемая к данным;

root - ранг процесса, получающего результат;

comm - коммуникатор, в рамках которого проходит операция редукции.