Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Дальневосточный федеральный университет»

(ДВФУ)

Капитан В.Ю.

Лабораторная работа 1

Процедуры блокирующего двухточечного обмена MPI

Владивосток

2022

**Двухточечные блокирующие обмены**

Участниками двухточечного обмена являются два процесса: процесс-отправитель и процесс-получатель. Блокирующие операции двухточечного обмена приостанавливают выполнение вызывающего процесса. Он переходит в состояние ожидания завершения передачи данных. Блокировка гарантирует выполнение действий в заданном порядке, обеспечивая предсказуемость поведения программы. С другой стороны, она создает условия для возникновения тупиковых ситуаций, когда оба процесса-участника обмена блокируются одновременно.

**Трансляция MPI-программ**

Утилиты трансляции и сборки находятся в каталоге /каталог\_установки\_MPI/bin. Его следует включить в путь поиска исполняемых файлов. Утилиты трансляции запускают трансляторы обычных языков C/C++ и Fortran. В командную строку их запуска, при необходимости, подставляют ссылки на необходимые библиотечные и заголовочные

файлы.

Для трансляции и компоновки программ на языке C++ используется команда mpiCC, для трансляции программ на языке C++ команда mpicc.

Информацию о ключах можно найти на справочных страницах MPICH (для их просмотра применяют команду man).

**Выполнение MPI-программ**

Для выполнения MPI-программ в MPICH используется загрузчик приложений mpirun.

Он запускает указанное количество копий программы. Команда запуска:

mpirun –np n [ключи MPI] название программы [ключи и аргументы программы]

где n - число запускаемых процессов.

**Лабораторная работа**

В заданиях лабораторной работы 1 нужно дописать и исправить исходные тексты на языке C, в том числе, процедуры блокирующего двухточечного обмена. Пропущенные фрагменты обозначены многоточием. Необходимо добавить эти вызовы, отладить, и запустить программы.

Необходимо предоставить отчет о лабораторной работе, содержащий полные, работоспособные исходные коды программ и скриншоты всего рабочего стола, с результатами выполнения ПО на кластере ДВФУ/

Задание 1.1

В исходном тексте программы отсутствуют команды подключения к MPI, а также определения количества процессов и ранга процесса.

#include "mpi.h"

#include <stdio.h>

int main(int argc,char \*argv[])

{

int my\_id, numprocs;

...

fprintf(stdout,"Process %d of %d\n", myid, numprocs);

MPI\_Finalize

return 0;

}

Задание 1.2

В исходном тексте программы на языке C отсутствуют команды подключения к MPI, а также команды блокирующего двухточечного обмена. Необходимо, чтобы при запуске двух процессов один из них отправлял сообщение второму.

#include "mpi.h"

#include <stdio.h>

int main(int argc,char \*argv[])

{

int myid, numprocs;

char message[20];

int murank;

MPI\_Status status;

...

if (myrank == 0)

{

strcpy(message, "Hi, Second Processor!");

... отправка

}

else

{

... получение

printf("received: %s\n", mesage);

}

MPI\_Finalize;

return 0;

}

Задание 1.3

Предполагается, что при запуске четного числа процессов, те из них, которые имеют четный ранг, отправляют сообщение следующим по величине ранга процессам. Реализовать программу на языке C/C++.