МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационные системы

Бариев Эмин Юсуфович

Институт информационных технологий и управления в технических системах

курс 4 группа ИС/б-16-2

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №1

По дисциплине: «ТРСиПВ»

По теме: «Иследование средств создания распределенно выполняющихся пограмм»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

ст.преподаватель Дрозин А. Ю.

(должность) (подпись) (инициалы,фамилия)

Севастополь 2019

1 ЦЕЛЬ РАБОТЫ

Исследовать функции библиотеки MPI, необходимые для создания и взаимодействия распределено выполняемых программ.

2 ПОСТАНОВКА ЗАДАЧИ

Вариант – 1

Программа осуществляет умножение двух матриц. Размеры матриц – 3\*4 и 4\*3. На каждом процессе, определяет произведение одной строки первой матрицы на все столбцы второй матрицы. Результаты возвращаются в родительскую задачу.

3 ХОД РАБОТЫ

3.1 Код программы на языке С++

#include <mpi.h>

#include <stdio.h>

#include <iostream>

#include <string>

**using** **namespace** std;

**int** main(**int** argc, **char**\*\* argv) {

**const** **int** n = 3;

**const** **int** m = 4;

**int** array1[n][m]={

{ 2 , 2 , 2, 2 },

{ 3 , 3 , 3, 3 },

{ 4 , 4 , 4, 4 },

};

**int** array2[m][n]={

{ 2 , 2 , 2 },

{ 3 , 3 , 3 },

{ 4 , 4 , 4 },

{ 5 , 5 , 5 },

};

**int** resultArray [n][n] = {

{0,0,0},

{0,0,0},

{0,0,0},

};

**int** tmlArr[n]= {0 , 0 , 0};

**int** res1[n]={0,0,0};

**int** res2[n]={0,0,0};

**int** res3[n]={0,0,0};

**int** rank;

**int** world;

MPI\_Status status;

MPI\_Init(&argc, &argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &world);

printf("Hello: rank %d, world: %d\n",rank, world);

**if** (rank == 0) {

///////////////SENDING//////////////////////////

MPI\_Send(\*array2, n\*m, MPI\_INT,1,99,MPI\_COMM\_WORLD);

MPI\_Send(array1[0], m, MPI\_INT,1,99,MPI\_COMM\_WORLD);

MPI\_Send(\*array2, n\*m, MPI\_INT,2,99,MPI\_COMM\_WORLD);

MPI\_Send(array1[1], m, MPI\_INT,2,99,MPI\_COMM\_WORLD);

MPI\_Send(\*array2, n\*m, MPI\_INT,3,99,MPI\_COMM\_WORLD);

MPI\_Send(array1[2], m, MPI\_INT,3,99,MPI\_COMM\_WORLD);

MPI\_Recv(tmlArr,n, MPI\_INT, 1, 98, MPI\_COMM\_WORLD, &status);

cout<<"get res from1"<<endl;

**for** (**int** i = 0; i<n; ++i) {

cout<<tmlArr[i]<<"\t";

resultArray[0][i] = tmlArr[i];

}

cout<<endl;

MPI\_Recv(tmlArr,n, MPI\_INT, 2, 98, MPI\_COMM\_WORLD, &status);

cout<<"get res from1"<<endl;

**for** (**int** i = 0; i<n; ++i) {

cout<<tmlArr[i]<<"\t";

resultArray[1][i] = tmlArr[i];

}

cout<<endl;

MPI\_Recv(tmlArr,n, MPI\_INT, 3, 98, MPI\_COMM\_WORLD, &status);

cout<<"get res from1"<<endl;

**for** (**int** i = 0; i<n; ++i) {

cout<<tmlArr[i]<<"\t";

resultArray[2][i] = tmlArr[i];

}

cout<<endl;

}

**if**(rank>0){

**for** (**int** r = 1; r<world; r++) {

MPI\_Recv(\*array2, n\*m, MPI\_INT, 0, 99, MPI\_COMM\_WORLD, &status);

MPI\_Recv(array1[r], m, MPI\_INT, 0, 99, MPI\_COMM\_WORLD, &status);

**int** temp = 0;

**for** (**int** k = 0; k<n; ++k) {

**for** (**int** i = 0; i<m; ++i) {

temp+=array1[r][i]\*array2[i][k];

}

tmlArr[k] = temp;

temp = 0;

}

MPI\_Send(tmlArr, n, MPI\_INT, 0, 98, MPI\_COMM\_WORLD);

}

}

cout<<endl;

cout<<"RESULTED MATRIX:"<<endl;

**for** (**int** i = 0; i<n; i++) {

**for** (**int** j = 0; j<n; j++) {

cout<<resultArray[i][j]<<"\t";

}

cout<<endl;

}

MPI\_Barrier(MPI\_COMM\_WORLD);

MPI\_Finalize();

**return** 0;

}

3.2 Результаты выполнения программы

**Hello: rank 0, world: 4**

**Hello: rank 1, world: 4**

**Hello: rank 2, world: 4**

**Hello: rank 3, world: 4**

**get res from1**

**28 28 28**

**get res from1**

**42 42 42**

**get res from1**

**56 56 56**

**RESULTED MATRIX:**

**28 28 28**

**42 42 42**

**56 56 56**

**Program ended with exit code: 9**

ВЫВОДЫ

В ходе выполнения лабораторной работы были изучены основные принципы и функции для работы с библиотекой MPI. Исследованы функции библиотеки MPI, необходимые для создания и взаимодействия распределено выполняемых программ.

Была реализована программа, осуществляющая умножение двух матриц размерностью 3x4 распределённо.