

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №00
по дисциплине «Параллельные алгоритмы»
Тема: Запуск параллельной программы на различном числе
одновременно работающих процессов, упорядочение вывода
результатов

Студентка гр. 9382

Иерусалимов Н.

Преподаватель

Татаринев Ю.С.

Санкт-Петербург

2021

Цель работы

Запустить программу с MPI, изучить работу программы.

Формулировка задания

Задание: Запустить программу на 1,2 ... N процессах несколько раз. Проанализировать порядок вывода сообщений на экран. Вывести правило, определяющее порядок вывода сообщений. Модифицировать программу таким образом, чтобы порядок вывода сообщений на экран соответствовал номеру соответствующего процесса.

```
#include <stdio.h>
#include "mpi.h"

int main(int argc, char* argv[]){
    int ProcNum, ProcRank, RecvRank;
    MPI_Status Status;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &ProcNum);
    MPI_Comm_rank(MPI_COMM_WORLD, &ProcRank);
    if ( ProcRank == 0 ){
        //Действия, выполняемые только процессом с рангом 0
        printf ("\n Hello from process %3d", ProcRank);
        for ( int i=1; i<ProcNum; i++ ) {
            MPI_Recv(&RecvRank, 1, MPI_INT, MPI_ANY_SOURCE,
                    MPI_ANY_TAG, MPI_COMM_WORLD, &Status);
            printf("\n Hello from process %3d", RecvRank);
        }
        printf("\n");
    }
    else // Сообщение, отправляемое всеми процессами,
        // кроме процесса с рангом 0
        MPI_Send(&ProcRank,1,MPI_INT,0,0, MPI_COMM_WORLD);
    MPI_Finalize();

    return 0;
}
```

Результат работы программы на 1,2 ... N процессорах

```
nereus@Nereus:~/CLionProjects/untitled1$ mpirun -np 1 ./a.out
Invalid MIT-MAGIC-COOKIE-1 key
Hello from process 0
nereus@Nereus:~/CLionProjects/untitled1$ mpirun -np 2 ./a.out
Invalid MIT-MAGIC-COOKIE-1 key
Hello from process 0
Hello from process 1
nereus@Nereus:~/CLionProjects/untitled1$ mpirun -np 3 ./a.out
Invalid MIT-MAGIC-COOKIE-1 key
Hello from process 0
Hello from process 1
Hello from process 2
nereus@Nereus:~/CLionProjects/untitled1$ mpirun -np 4 ./a.out
Invalid MIT-MAGIC-COOKIE-1 key
Hello from process 0
Hello from process 3
Hello from process 1
Hello from process 2
nereus@Nereus:~/CLionProjects/untitled1$ mpirun -np 5 ./a.out
Invalid MIT-MAGIC-COOKIE-1 key
Hello from process 0
Hello from process 4
Hello from process 1
Hello from process 2
Hello from process 3
nereus@Nereus:~/CLionProjects/untitled1$ mpirun -np 6 ./a.out
Invalid MIT-MAGIC-COOKIE-1 key
Hello from process 0
Hello from process 5
Hello from process 3
Hello from process 1
Hello from process 4
Hello from process 2
nereus@Nereus:~/CLionProjects/untitled1$ mpirun -np 7 ./a.out
Invalid MIT-MAGIC-COOKIE-1 key
Hello from process 0
Hello from process 5
Hello from process 3
Hello from process 1
Hello from process 6
Hello from process 2
```

```
Hello from process 4
```

Анализ порядка вывода сообщений на экран

Номер процесса – его ранг. ProcRank отвечает за хранение ранга процесса, выполнившего вызов. Пересылка сообщения принимается процессом-получателем при помощи функции MPI_Recv. С помощью него в нашей программе сообщения принимаются от любого процесса-отправителя с помощью MPI_ANY_SOURCE. Все сообщения начинают выводиться только после того, как ранг текущего процесса станет 0, печатается значение ранга текущего процесса, затем выполняется цикл для получения сообщений и их отображения в консоли. Порядок сообщений не определен заранее, так как зависит от условий выполнения параллельной программы.

Модификация программы

```
#include <stdio.h>
#include "mpi.h"
int main(int argc, char* argv[]) {
    int ProcNum, ProcRank, RecvRank;
    MPI_Status Status;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &ProcNum);
    MPI_Comm_rank(MPI_COMM_WORLD, &ProcRank);
    if (ProcRank == 0) {
        //Действия, выполняемые только процессом с рангом 0
        printf("\n Hello from process %3d", ProcRank);
        for (int i = 1; i < ProcNum; i++) {
            MPI_Recv(&RecvRank, 1, MPI_INT, i,
                    MPI_ANY_TAG, MPI_COMM_WORLD, &Status);
            printf("\n Hello from process %3d", RecvRank);
        }
        printf("\n");
    }
    else // Сообщение, отправляемое всеми процессами,
        // кроме процесса с рангом 0
        MPI_Send(&ProcRank, 1, MPI_INT, 0, 0, MPI_COMM_WORLD);
    MPI_Finalize();
}
```

```
return 0;  
}
```

Результат вывода модифицированной программы:

```
nereus@Nereus:~/CLionProjects/untitled1$ mpirun -np 8 ./a.out  
Invalid MIT-MAGIC-COOKIE-1 key  
Hello from process 0  
Hello from process 1  
Hello from process 2  
Hello from process 3  
Hello from process 4  
Hello from process 5  
Hello from process 6  
Hello from process 7  
nereus@Nereus:~/CLionProjects/untitled1$
```

Для того чтобы программа выводила сообщения по порядку принимающая функция должна принимать сообщения по порядку в цикле вместо `MPI_ANY_SOURCE`. Так как порядок строго определен, работа параллельных вычислений может быть замедлена.

Выводы

В ходе лабораторной работы была проанализирована и модифицирована данная программа MPI, были изучены базовые функции MPI.