

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

ОТЧЕТ
по лабораторной работе №01
по дисциплине «ПА»
Тема: Обмен сообщениями чётных и нечётных процессов

Студент гр. 8304

Сергеев А.Д.

Преподаватель

Татаринов Ю.С.

Санкт-Петербург

2020

Задание.

Напишите программу обмена сообщениями чётных и нечётных процессов. Замерьте время на одну итерацию обмена и определите зависимость времени обмена от длины сообщения.

Описание алгоритма.

Нулевой процесс создаёт массив байтов заданного размера, замеряет текущее время и передаёт массив первому процессу. После чего ожидает сообщения, содержащего тот же массив от последнего процесса. После чего он второй раз замеряет время и выводит на экран разницу. Среднее время передачи сообщения может быть получено делением общего времени передачи на количество процессов.

Каждый из ненулевых процессов ожидает сообщения, содержащего массив, от предыдущего процесса, после получения которого выводит на экран сообщение, содержащее текущее время и посылает массив следующему процессу.

Листинг программы.

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

#define size 100

int main(int argc, char* argv[]) {
    int ProcNum, ProcRank;
    MPI_Status Status;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &ProcNum);
    MPI_Comm_rank(MPI_COMM_WORLD, &ProcRank);

    char* bytes = malloc(size * sizeof(char));

    if (ProcRank == 0) {
        printf("Processes exchange blocks, %d bytes each\n", size);
        double start_time = MPI_Wtime();
        printf("Root process started at %lf\n", start_time);
```

```

MPI_Send(bytes, size, MPI_BYTE, 1, ProcRank, MPI_COMM_WORLD);
MPI_Recv(bytes, size, MPI_BYTE, MPI_ANY_SOURCE, ProcNum - 1,
MPI_COMM_WORLD, &Status);

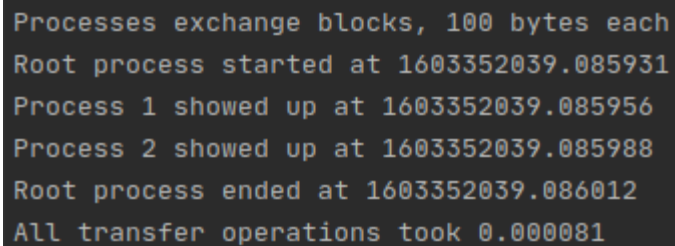
double end_time = MPI_Wtime();
printf("Root process ended at %lf\n", end_time);
printf("All transfer operations took %lf\n", end_time - start_time);

} else {
MPI_Recv(bytes, size, MPI_BYTE, MPI_ANY_SOURCE, ProcRank - 1,
MPI_COMM_WORLD, &Status);
printf("Process %d showed up at %lf\n", ProcRank, MPI_Wtime());
int dest = ProcRank == ProcNum - 1 ? 0 : ProcRank + 1;
MPI_Send(bytes, size, MPI_BYTE, dest, ProcRank, MPI_COMM_WORLD);
}

free(bytes);
MPI_Finalize();
return 0;
}

```

Результаты работы.



```

Processes exchange blocks, 100 bytes each
Root process started at 1603352039.085931
Process 1 showed up at 1603352039.085956
Process 2 showed up at 1603352039.085988
Root process ended at 1603352039.086012
All transfer operations took 0.000081

```

Рисунок 1 - Результат работы программы для массива из 100 элементов и трёх процессов

```
Processes exchange blocks, 100 bytes each
Root process started at 1603352066.796661
Process 1 showed up at 1603352066.800445
Process 2 showed up at 1603352066.808469
Process 3 showed up at 1603352066.816444
Process 4 showed up at 1603352066.816615
Process 5 showed up at 1603352066.819461
Process 6 showed up at 1603352066.820940
Process 7 showed up at 1603352066.828428
Process 8 showed up at 1603352066.828455
Process 9 showed up at 1603352066.840433
Process 10 showed up at 1603352066.848432
Root process ended at 1603352066.848532
All transfer operations took 0.051872
```

Рисунок 2 - Результат работы программы для массива из 100 элементов и одиннадцати процессов

```
Processes exchange blocks, 100 bytes each
Root process started at 1603352160.914872
Process 1 showed up at 1603352160.950625
Process 2 showed up at 1603352160.951122
Process 3 showed up at 1603352160.972435
Process 4 showed up at 1603352160.972468
Process 5 showed up at 1603352160.980928
Process 6 showed up at 1603352160.995326
Process 7 showed up at 1603352160.995358
Process 8 showed up at 1603352160.996436
Process 9 showed up at 1603352161.001965
Process 10 showed up at 1603352161.020528
Process 11 showed up at 1603352161.040436
Process 12 showed up at 1603352161.060458
Process 13 showed up at 1603352161.077081
Process 14 showed up at 1603352161.091926
Process 15 showed up at 1603352161.098459
Process 16 showed up at 1603352161.108026
Process 17 showed up at 1603352161.108437
Process 18 showed up at 1603352161.114545
Process 19 showed up at 1603352161.120441
Process 20 showed up at 1603352161.134530
Process 21 showed up at 1603352161.138578
Process 22 showed up at 1603352161.138732
Root process ended at 1603352161.152435
All transfer operations took 0.237563
```

Рисунок 3 - Результат работы программы для массива из 100 элементов и двадцати трёх процессов

```
Processes exchange blocks, 10000 bytes each
Root process started at 1603352200.558996
Process 1 showed up at 1603352200.559065
Process 2 showed up at 1603352200.559314
Root process ended at 1603352200.559439
All transfer operations took 0.000443
```

Рисунок 4 - Результат работы программы для массива из 10000 элементов
и трёх процессов

```
Processes exchange blocks, 10000 bytes each
Root process started at 1603352229.529477
Process 1 showed up at 1603352229.532443
Process 2 showed up at 1603352229.553422
Process 3 showed up at 1603352229.560216
Process 4 showed up at 1603352229.565373
Process 5 showed up at 1603352229.572445
Process 6 showed up at 1603352229.577818
Process 7 showed up at 1603352229.579289
Process 8 showed up at 1603352229.590061
Process 9 showed up at 1603352229.605655
Process 10 showed up at 1603352229.624465
Root process ended at 1603352229.644447
All transfer operations took 0.114969
```

Рисунок 5 - Результат работы программы для массива из 10000 элементов
и одиннадцати процессов

```
Processes exchange blocks, 10000 bytes each
Root process started at 1603352251.440418
Process 1 showed up at 1603352251.472451
Process 2 showed up at 1603352251.496442
Process 3 showed up at 1603352251.508537
Process 4 showed up at 1603352251.528444
Process 5 showed up at 1603352251.532443
Process 6 showed up at 1603352251.543206
Process 7 showed up at 1603352251.552444
Process 8 showed up at 1603352251.572472
Process 9 showed up at 1603352251.605547
Process 10 showed up at 1603352251.616441
Process 11 showed up at 1603352251.620862
Process 12 showed up at 1603352251.636455
Process 13 showed up at 1603352251.640437
Process 14 showed up at 1603352251.646540
Process 15 showed up at 1603352251.652446
Process 16 showed up at 1603352251.676448
Process 17 showed up at 1603352251.696440
Process 18 showed up at 1603352251.709897
Process 19 showed up at 1603352251.718210
Process 20 showed up at 1603352251.720637
Process 21 showed up at 1603352251.720722
Process 22 showed up at 1603352251.720993
Root process ended at 1603352251.728435
All transfer operations took 0.288017
```

Рисунок 6 - Результат работы программы для массива из 10000 элементов и двадцати трёх процессов

```
Processes exchange blocks, 1000000 bytes each
Root process started at 1603352283.139770
Process 1 showed up at 1603352283.141527
Process 2 showed up at 1603352283.143000
Root process ended at 1603352283.144580
All transfer operations took 0.004810
```

Рисунок 7 - Результат работы программы для массива из 1000000 элементов и трёх процессов

```
Processes exchange blocks, 1000000 bytes each
Root process started at 1603352309.896838
Process 1 showed up at 1603352309.906474
Process 2 showed up at 1603352309.909472
Process 3 showed up at 1603352309.921993
Process 4 showed up at 1603352309.992905
Process 5 showed up at 1603352309.998668
Process 6 showed up at 1603352310.010037
Process 7 showed up at 1603352310.024084
Process 8 showed up at 1603352310.039028
Process 9 showed up at 1603352310.061205
Process 10 showed up at 1603352310.089057
Root process ended at 1603352310.105322
All transfer operations took 0.208484
```

Рисунок 8 - Результат работы программы для массива из 1000000 элементов и одиннадцати процессов

```
Processes exchange blocks, 1000000 bytes each
Root process started at 1603352328.963877
Process 1 showed up at 1603352328.989562
Process 2 showed up at 1603352329.010396
Process 3 showed up at 1603352329.114295
Process 4 showed up at 1603352329.216818
Process 5 showed up at 1603352329.233144
Process 6 showed up at 1603352329.296906
Process 7 showed up at 1603352329.425651
Process 8 showed up at 1603352329.599160
Process 9 showed up at 1603352329.639367
Process 10 showed up at 1603352329.760579
Process 11 showed up at 1603352329.873565
Process 12 showed up at 1603352329.881243
Process 13 showed up at 1603352329.984372
Process 14 showed up at 1603352330.013200
Process 15 showed up at 1603352330.160560
Process 16 showed up at 1603352330.212391
Process 17 showed up at 1603352330.352567
Process 18 showed up at 1603352330.353866
Process 19 showed up at 1603352330.381183
Process 20 showed up at 1603352330.507890
Process 21 showed up at 1603352330.544895
Process 22 showed up at 1603352330.684566
Root process ended at 1603352330.705212
All transfer operations took 1.741335
```

Рисунок 9 - Результат работы программы для массива из 1000000 элементов и двадцати трёх процессов

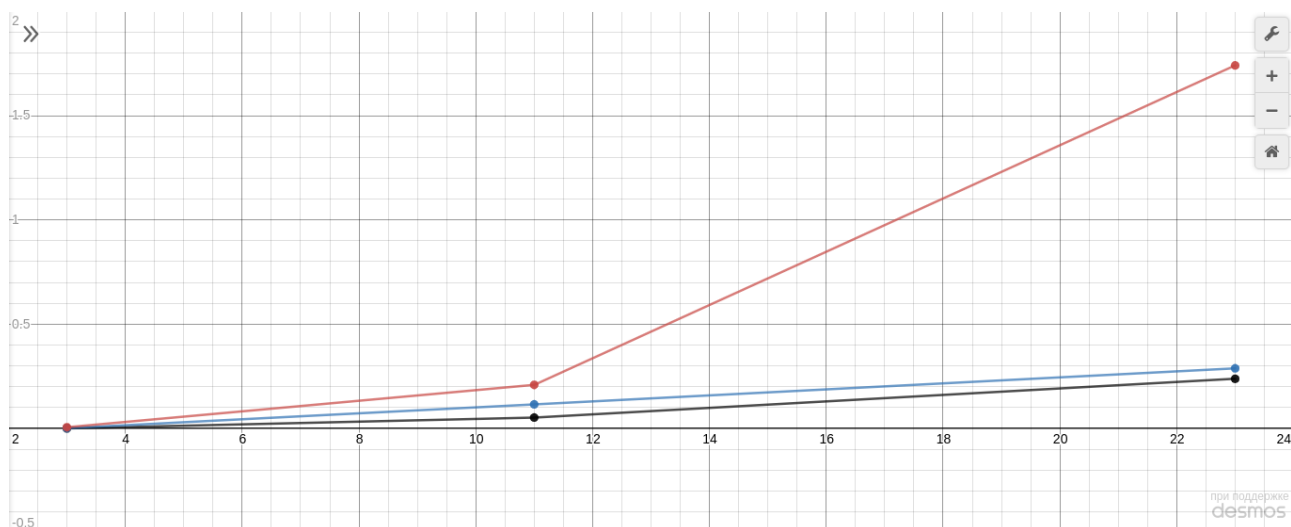


Рисунок 10 — Графики зависимости общего времени передачи сообщений от количества процессов.

Черный график — массив из 100 элементов.

Синий график — массив из 10000 элементов.

Красный график — массив из 1000000 элементов.

Выводы.

Написана программа, измеряющая зависимость времени передачи сообщений между процессами от их длины. Полученные данные демонстрируют сравнительно быструю скорость передачи для большого количества данных (~ 1 мб) и практически идентичную скорость для данных в пределах 10 кб. Это говорит о хорошей оптимизации данной реализации MPI.