FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER EDUCATION

ITMO UNIVERSITY

Report

MPI. Assignments 6 – 8

Parallel algorithms for the analysis and synthesis of data

Performed by
Aleksandr Shirokov
J4133c
Accepted by
Petr Andriushchenko
Deadline: 20.12.21

St. Petersburg
2021

Contents

1 Assignments

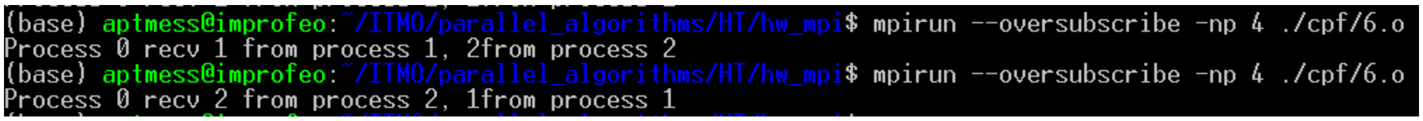
1.1 Assignment 6.

1.1.1 Formulation of the problem

1. Compile the example ASSIGNMENT6.C in detail, run it and explain it.
2. Transform the program using the MPI_TAG field of the status structure in the condition.

1.1.2 Example of launch parameters and output. Detailed description of solution

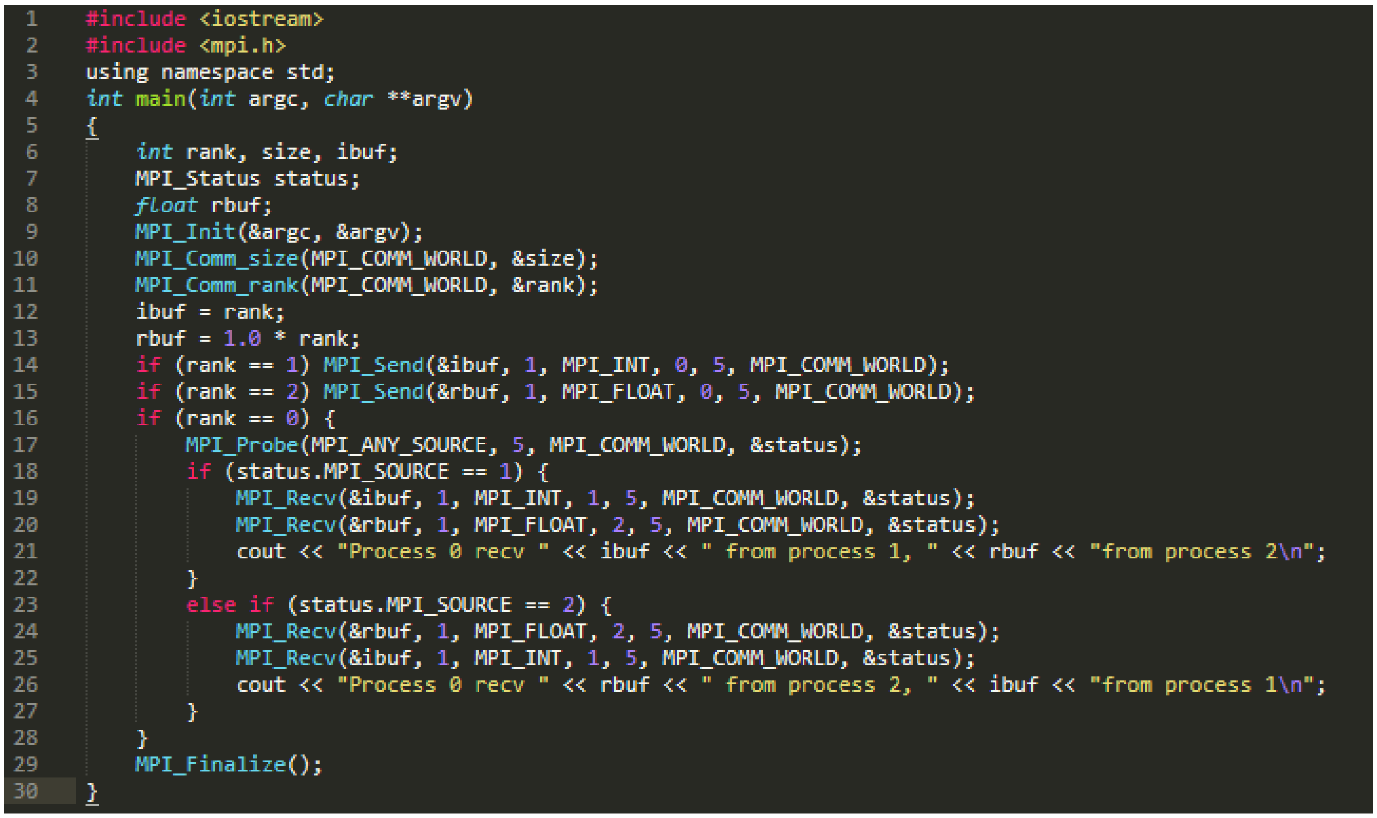
Code for **assignment 6** is here.
Compilation example: MPIC++ -O ./CPF/6.O ASSIGNMENT6.C
Launch example: MPIRUN –OVERSUBSCRIBE -NP 4 ./CPF/6.O



```
(base) aptmess@improfeo:~/ITMO/parallel_algorithms/HT/hw_mpi$ mpirun --oversubscribe -np 4 ./cpf/6.o
Process 0 recv 1 from process 1, 2from process 2
(base) aptmess@improfeo:~/ITMO/parallel_algorithms/HT/hw_mpi$ mpirun --oversubscribe -np 4 ./cpf/6.o
Process 0 recv 2 from process 2, 1from process 1
```

There could be only two results of program output

Let's move to the the code and explain how it works.



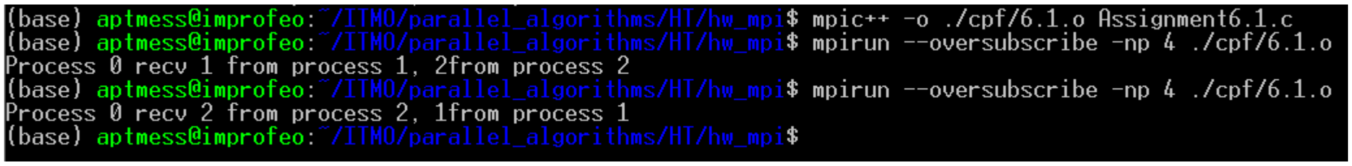
```
#include <iostream>
#include <mpi.h>
using namespace std;
int main(int argc, char **argv)
{
    int rank, size, ibuf;
    MPI_Status status;
    float rbuf;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    ibuf = rank;
    rbuf = 1.0 * rank;
    if (rank == 1) MPI_Send(&ibuf, 1, MPI_INT, 0, 5, MPI_COMM_WORLD);
    if (rank == 2) MPI_Send(&rbuf, 1, MPI_FLOAT, 0, 5, MPI_COMM_WORLD);
    if (rank == 0) {
        MPI_Probe(MPI_ANY_SOURCE, 5, MPI_COMM_WORLD, &status);
        if (status.MPI_SOURCE == 1) {
            MPI_Recv(&ibuf, 1, MPI_INT, 1, 5, MPI_COMM_WORLD, &status);
            MPI_Recv(&rbuf, 1, MPI_FLOAT, 2, 5, MPI_COMM_WORLD, &status);
            cout << "Process 0 recv " << ibuf << " from process 1, " << rbuf << "from process 2\n";
        }
        else if (status.MPI_SOURCE == 2) {
            MPI_Recv(&rbuf, 1, MPI_FLOAT, 2, 5, MPI_COMM_WORLD, &status);
            MPI_Recv(&ibuf, 1, MPI_INT, 1, 5, MPI_COMM_WORLD, &status);
            cout << "Process 0 recv " << rbuf << " from process 2, " << ibuf << "from process 1\n";
        }
    }
    MPI_Finalize();
}
```

Assignment6 code

Firstly there is an initialization of parallel part using MPI_Init, after if rank of process is 1 then the int 1 will be send as a message and if rank of process is 2, then the float value 2.0 will be send as message. After we are going to main process 0 logic:

- MPI_Probe this function is waiting for message from any process with msgtag = 5 and wouldn't go next if the message doesn't come to process 0. Let's make it clear - function only understand that message come to process, but doesn't get it.
- After that if status.MPI_SOURCE == 1 so if first was message from process 1 then there is a print message that 1st process's message was quicklier, else - that the second was quicklier and the value from second process will be displayed first.

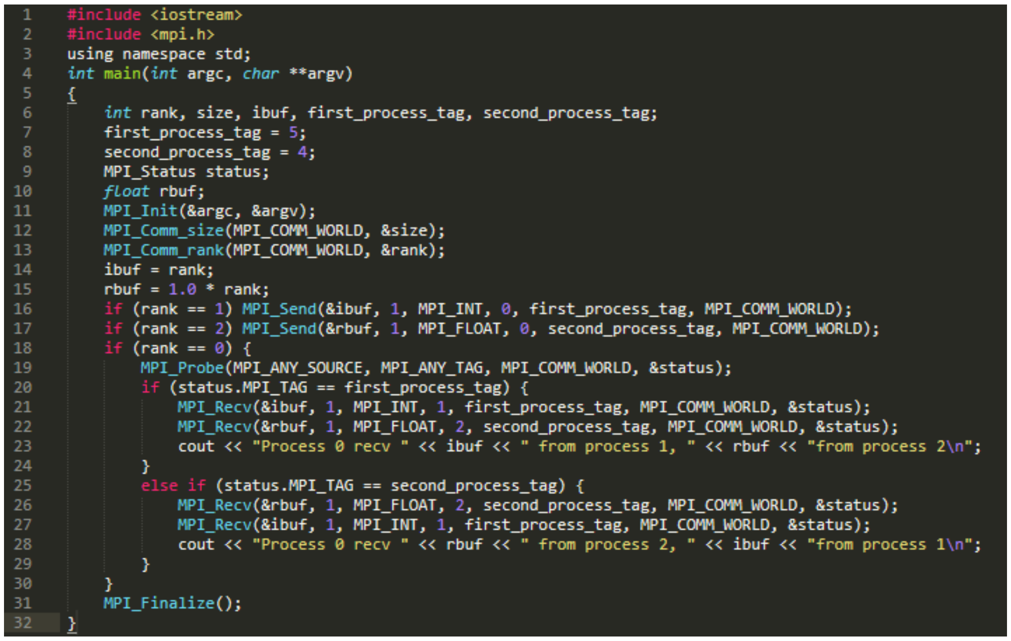
After I have transformed the problem using MPI_TAG field. Here are results:



```
(base) aptmess@improfeo:~/ITMO/parallel_algorithms/HT/hw_mpi$ mpic++ -o ./cpf/6.1.o Assignment6.1.c
(base) aptmess@improfeo:~/ITMO/parallel_algorithms/HT/hw_mpi$ mpirun --oversubscribe -np 4 ./cpf/6.1.o
Process 0 recv 1 from process 1, 2from process 2
(base) aptmess@improfeo:~/ITMO/parallel_algorithms/HT/hw_mpi$ mpirun --oversubscribe -np 4 ./cpf/6.1.o
Process 0 recv 2 from process 2, 1from process 1
(base) aptmess@improfeo:~/ITMO/parallel_algorithms/HT/hw_mpi$
```

Results are the same. Take a look at code

Code for **assignment 6.1** is here.
Compilation example: mpic++ -o ./cpf/6.1.o Assignment6.1.c
Launch example: mpirun –oversubscribe -np 4 ./cpf/6.1.o



```cpp
#include <iostream>
#include <mpi.h>
using namespace std;
int main(int argc, char **argv)
{
    int rank, size, ibuf, first_process_tag, second_process_tag;
    first_process_tag = 5;
    second_process_tag = 4;
    MPI_Status status;
    float rbuf;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    ibuf = rank;
    rbuf = 1.0 * rank;
    if (rank == 1) MPI_Send(&ibuf, 1, MPI_INT, 0, first_process_tag, MPI_COMM_WORLD);
    if (rank == 2) MPI_Send(&rbuf, 1, MPI_FLOAT, 0, second_process_tag, MPI_COMM_WORLD);
    if (rank == 0) {
        MPI_Probe(MPI_ANY_SOURCE, MPI_ANY_TAG, MPI_COMM_WORLD, &status);
        if (status.MPI_TAG == first_process_tag) {
            MPI_Recv(&ibuf, 1, MPI_INT, 1, first_process_tag, MPI_COMM_WORLD, &status);
            MPI_Recv(&rbuf, 1, MPI_FLOAT, 2, second_process_tag, MPI_COMM_WORLD, &status);
            cout << "Process 0 recv " << ibuf << " from process 1, " << rbuf << "from process 2\n";
        }
        else if (status.MPI_TAG == second_process_tag) {
            MPI_Recv(&rbuf, 1, MPI_FLOAT, 2, second_process_tag, MPI_COMM_WORLD, &status);
            MPI_Recv(&ibuf, 1, MPI_INT, 1, first_process_tag, MPI_COMM_WORLD, &status);
            cout << "Process 0 recv " << rbuf << " from process 2, " << ibuf << "from process 1\n";
        }
    }
    MPI_Finalize();
}
```

Assignment6 part II code

Everything is more or less the same, but now we are expecting any tag in MPI_Probe function and processes 1 and 2 has different tags (5 and 4) and condition is also have changed (status.MPI_TAG). Program works correctly.

1.2 Assignment 7.

1.2.1 Formulation of the problem

1.2.2 Example of launch parameters and output. Detailed description of solution

Code for **assignment 7** is here.
Compilation example: mpic++ -o ./cpf/7.o Assignment7.c
Launch example: mpirun –oversubscribe -np 4 ./cpf/7.o
Let's move to the the code and explain how it works.
Explain.

1.3 Assignment 8.

1.3.1 Formulation of the problem

1.3.2 Example of launch parameters and output. Detailed description of solution

Code for **assignment 8** is here.
Compilation example: mpic++ -o ./cpf/6.o Assignment6.c
Launch example: mpirun –oversubscribe -np 4 ./cpf/6.o
Let's move to the the code and explain how it works.
Explain.

1.4 Appendix

The link to the sourse code which is placed on my github.