Saint Petersburg National Research University of Information Technologies, Mechanics and Optics (ITMO University)

Report

**about laboratory works**

**Assignment 16.**

**Assignment 17.**

**Assignment 18.**

**Assignment 19.**

**Assignment 20.**

**Assignment 21.**

**Student** Pogrebnoy D.A. j4132c

Saint-Petersburg, 2021

# Assignment 16.

**Task**

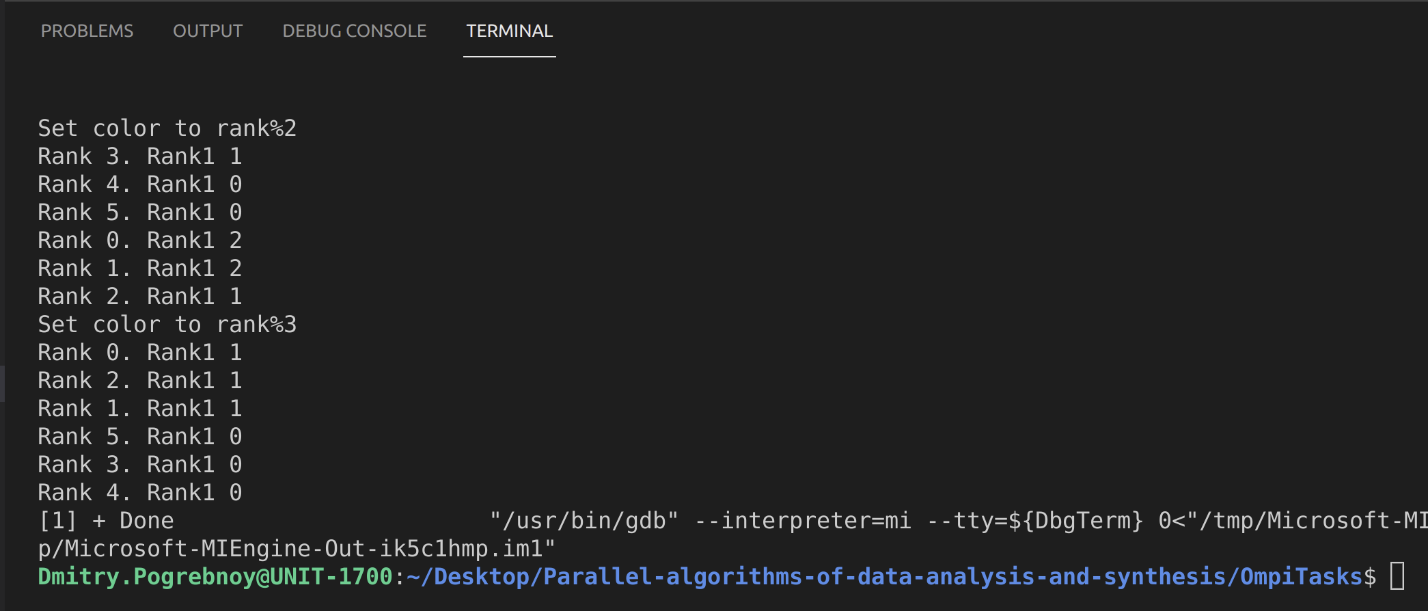
In the MPI\_Comm\_split function (Assignment16.c), replace the color parameter with (rank% 2), (rank% 3), look at how many groups the processes are split into, depending on the specified attribute of division into groups.

**Implementation**

Source code and data gathered are available on <https://github.com/DmitryPogrebnoy/Parallel-algorithms-of-data-analysis-and-synthesis/blob/master/OmpiTasks/Task16/Assignment16.cpp>

The description of the code is described in the comments.

Output example:



# Assigments 17.

**Task**

Understand the new functions in Assignment17.c. and explain program execution.

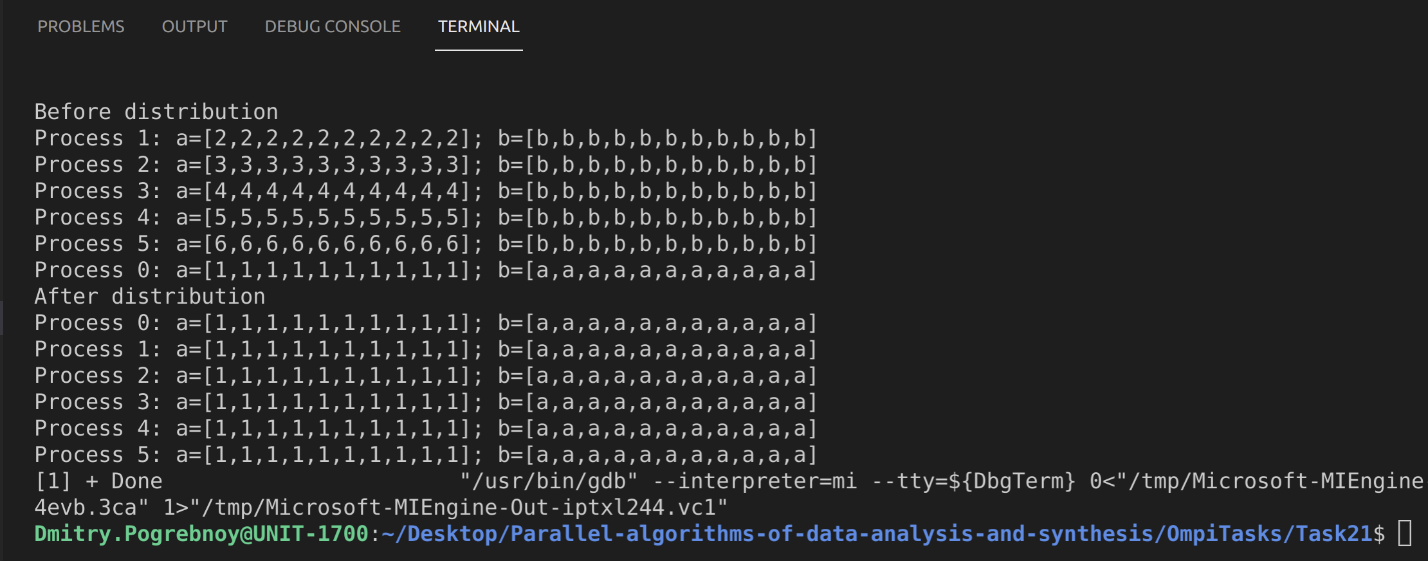
Display the values of the process number and arrays a[i], b[i], before packing and distribution, and after. See how broadcasting works.

**Implementation**

Source code and data gathered are available on <https://github.com/DmitryPogrebnoy/Parallel-algorithms-of-data-analysis-and-synthesis/blob/master/OmpiTasks/Task17/Assignment17.cpp>

The description of the code is described in the comments.

Output example:



# Assignment 18.

**Task**

To complete the task, you need to create and compile two programs: Master (master.o) and Slave (slave.o). The Master should start the worker, so be careful with the names of the executable files.

Launch the master via the mpiexec command for one process.

Startup example: mpiexec -n 1 ./master.o

Understand the new functions in Assignment18\_master.c and Assignment18\_slave.c and explain programs execution.

Add a third process, which will transfer from the slave processes to the master the number of running processes, the master should receive and display.

**Implementation**

Source code and data gathered are available on <https://github.com/DmitryPogrebnoy/Parallel-algorithms-of-data-analysis-and-synthesis/tree/master/OmpiTasks/Task18>

The description of the code is described in the comments.

Output of implemented task example:



# Assignment 19.

**Task**

To complete the task, you need to create and compile two programs: server and client. In one window of the SSH client, a server is launched for one process, which gives out the port name.

An example of a command to start the server: mpiexec -n 1 ./serv.o

Then the client is launched in another window, specifying the port name separated by a space in single quotes (example command: mpiexec -n 1 ./client.o ‘port name’).

Understand the new functions in Assignment19\_serv.c and Assignment19\_client.c and explain programs execution.

Check the work by running the server and the client. Add the program and send an arbitrary message to each other.

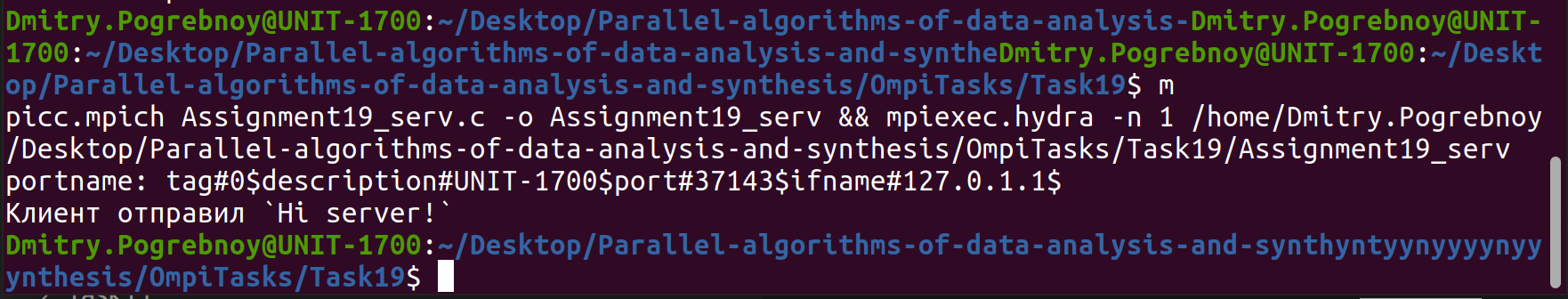
**Implementation**

Source code and data gathered are available on <https://github.com/DmitryPogrebnoy/Parallel-algorithms-of-data-analysis-and-synthesis/tree/master/OmpiTasks/Task19>

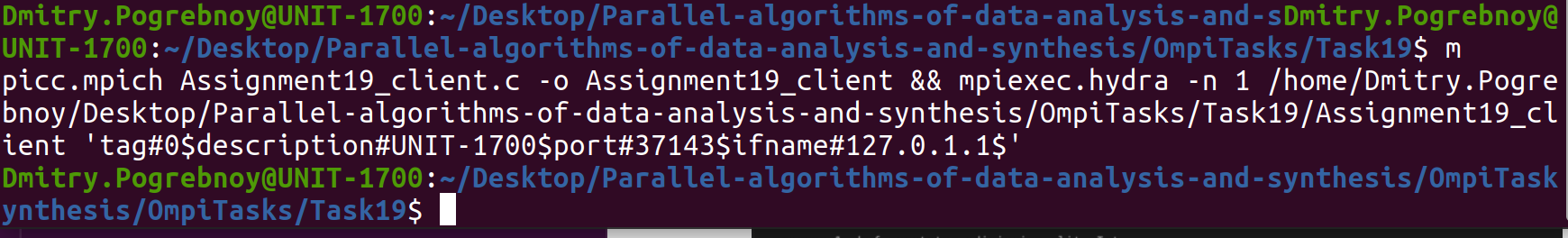
The description of the code is described in the comments.

OpenMPI is broken, so I used MPICH to compile and run the source code.

Run server:



Run client:



# Assignment 20.

**Task**

Understand the new functions in Assignment20.c, complete the program according to the assignment, explain the execution of the program.

Write a function that will create a file "file.txt" with random content (or with specific text). The function must be executed before the program reads the contents of the file. Run the program on one process. Check if the contents of the file are displayed correctly. Add an option that will delete the file on close.

**Implementation**

Source code and data gathered are available on <https://github.com/DmitryPogrebnoy/Parallel-algorithms-of-data-analysis-and-synthesis/blob/master/OmpiTasks/Task20/Assignment20.c>

The description of the code is described in the comments.

Output of implemented task example:



# Assignment 21.

**Task**

Understand the new functions in Assignment21.c, complete the program according to the assignment, explain the execution of the program.

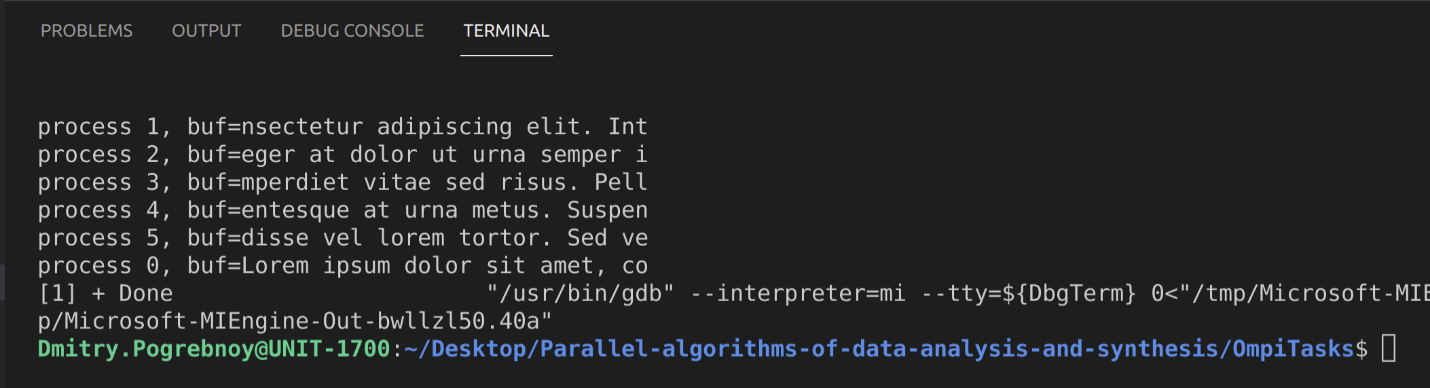
Create a file and fill it with bulky text, output the content in parallel. Change the step of reading the contents of the file and the number of characters to be output by each process.

**Implementation**

Source code and data gathered are available on <https://github.com/DmitryPogrebnoy/Parallel-algorithms-of-data-analysis-and-synthesis/tree/master/OmpiTasks/Task21>

The description of the code is described in the comments.

Output of implemented task example:



And file contains following data:

