**MPI**

**Лабораторная работа №1**

Милосердова Лидия ПИН-34

Код программы:

#include <stdio.h>

#include <mpi.h>

#include <iostream>

#include <Windows.h>

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

{

//Variables init

int numtasks, rank;

int mes = 0, procNum = 0;

bool finished = false;

MPI\_Status status;

srand(time(NULL));

//MPI starts

MPI\_Init(argc, &argv);

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

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

//Main proccess show how many tasks used

if (rank == 0)

{

std::cout << "Num of all tasks: " << numtasks << std::endl;

}

//Proccesses cycle

while (!finished)

{

//Main proccess part

if (rank == 0)

{

static int counter = 0;

MPI\_Recv(&mes, 1, MPI\_INT, MPI\_ANY\_SOURCE, MPI\_ANY\_TAG, MPI\_COMM\_WORLD, &status);

//finish counter

if (mes == -1)

{

std::cout << "Proc " << rank << " Get: " << mes << " From: " << status.MPI\_SOURCE << " Counter: " << counter <<" Proc finished" << std::endl;

finished = true;

//rend reply to kill proccess

for (int i = 1; i < numtasks; i++)

MPI\_Send(&mes, 1, MPI\_INT, i, 0, MPI\_COMM\_WORLD);

}

//Increment and show value of counter

else

{

counter++;

std::cout << "Proc " << rank << " Get: " << mes <<" From: " << status.MPI\_SOURCE << " Counter: " << counter << std::endl;

MPI\_Send(&mes, 1, MPI\_INT, status.MPI\_SOURCE, 0, MPI\_COMM\_WORLD);

}

}

//Other proccesses part

else

{

//random message

mes = rand() % 10 - 1;

//send to counter

MPI\_Send(&mes, 1, MPI\_INT, 0, 0, MPI\_COMM\_WORLD);

//wait reply

MPI\_Recv(&mes, 1, MPI\_INT, 0, MPI\_ANY\_TAG, MPI\_COMM\_WORLD, &status);

//if reply == -1 finish proccess

if (mes == -1)

{

finished = true;

std::cout << "Proc " << rank << " finished" << std::endl;

}

//delay

Sleep(1);

}

}

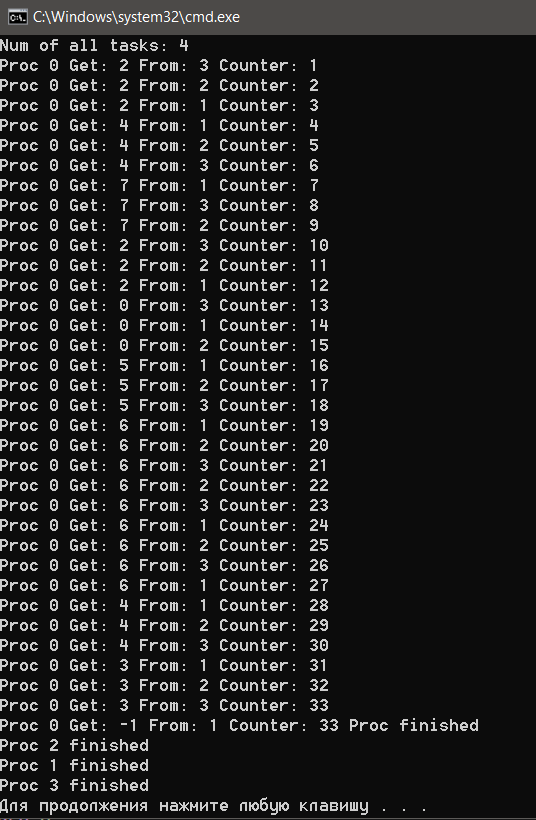
MPI\_Finalize();

return 0;

}

Скриншоты выполнения:

1. 4 процесса (3 отправителя, 1 счетчик)



1. 20 процессов (19 отправителей, 1 счетчик)

