**Федеральное государственное бюджетное образовательное учреждение высшего образования «Ивановский государственный энергетический университет имени В.И. Ленина»**

**Отчет по MPI\_Bank**

**Выполнил: Тютюкин Е.С.**

**Проверила: Чернышева Л.П.**

**Иваново 2020**

**Используемые функции:**

**int** MPI\_Send (**void** \*buf, **int** count, MPI\_Datatype type, **int** dest, **int** tag, MPI\_Comm comm)

buf - адрес буфера памяти, в котором располагаются данные отправляемого сообщения,

count - количество элементов данных пересылаемого сообщения,

type - тип элементов данных пересылаемого сообщения,

dest - ранг процесса, которому отправляется сообщение,

tag - значение-тег, используемое для идентификации сообщения,

comm - коммуникатор, в рамках которого выполняется передача данных.

**int** MPI\_Recv (**void** \*buf, **int** count, MPI\_Datatype type, **int** source, **int** tag, MPI\_Comm comm, MPI\_Status \*status)

buf - адрес буфера памяти, в котором располагаются данные принимаемого сообщения,

count - количество элементов данных принимаемого сообщения;

type - тип элементов данных принимаемого сообщения;

source - ранг процесса, от которого должен быть выполнен прием сообщения;

tag - тег сообщения, которое должно быть принято для процесса;

comm - коммуникатор, в рамках которого выполняется передача данных;

status - указатель на структуру данных с информацией о результате выполнения.

**Текст программы со стратегией:**

//#include "pch.h"

#include <iostream>

#include <mpi.h>

#include <stdlib.h>

using namespace std;

int main(int arc, char\*\* argv) {

int size, rank, msgtag = 12;

MPI\_Status status;

//шапочка

if (MPI\_Init(&arc, &argv) != MPI\_SUCCESS) return 1;

if (MPI\_Comm\_size(MPI\_COMM\_WORLD, &size) != MPI\_SUCCESS) {

MPI\_Finalize();

return 2;

}

if (MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank) != MPI\_SUCCESS) {

MPI\_Finalize();

return 3;

}

// cout << "my rank: " << rank << endl;

#define comm MPI\_COMM\_WORLD

#define lmax 20 //жизненный цикл

#define omax 10 //

#define n 2

int pr = 1, mag[omax \* n], balance, z[n];

int\* p1, \* nmag, m, kmax, sum, nz = 10;

p1 = nmag = mag;

FILE\* fres;

char name[20];

sprintf\_s(name, "result%d.txt", rank);

fopen\_s(&fres, name, "w");

if (!rank) //банк

{

balance = 100;

kmax = size - 1;

}

if (rank) //клиент

{

z[0] = rand() % 2;

z[1] = (rand() % 100) \* rank + 1;

fprintf(fres, "клиент %d имеет статус z[0] %d и сумму z[1] %d\n", rank, z[0], z[1]);

}

fclose(fres);

MPI\_Barrier(MPI\_COMM\_WORLD);

for (int l = 0; l < lmax; l++) //жизненный цикл

{

fopen\_s(&fres, name, "a");

if (!rank) //банк

{

for (int i = 0; i < kmax; i++) //до мах клиента

{

MPI\_Recv(z, n, MPI\_INT, MPI\_ANY\_SOURCE, msgtag, comm, &status);

if (z[0]) //=1 если кладем деньги

{

balance += z[1];

pr = 1;

MPI\_Send(&pr, 1, MPI\_INT, status.MPI\_SOURCE, msgtag, comm);

fprintf(fres, "клиент %d PUT = %d$, Текущий balance = %d\n", status.MPI\_SOURCE, z[1], balance);

}

else //z[0]=0 если снимаем

{

if (z[1] > balance - nz) //помещаем в очередь

{

fprintf(fres, "в очереди: клиент %d имеет сумму z[1] %d\n", status.MPI\_SOURCE, z[1]);

\*p1 = status.MPI\_SOURCE; //указатель на клиента

p1++;

\*p1 = z[1];

p1++;

}

else

{

balance -= z[1];

pr = 1;

MPI\_Send(&pr, 1, MPI\_INT, status.MPI\_SOURCE, msgtag, comm);

fprintf(fres, "банк выдает клиенту %d сумму z[1] = %d Текущий balance = %d\n", status.MPI\_SOURCE, z[1], balance);

}

}

}

int\* p2 = nmag;

while (p1 != p2) //пока очередь не пустая

{

m = \*p2; //кому

sum = \*(p2 + 1);

if (sum < balance - nz) //можно выдать

{

balance -= sum;

pr = 1;

MPI\_Send(&pr, 1, MPI\_INT, m, msgtag, comm);

fprintf(fres, "банк может выдать %d клиенту сумму %d, текущий balance=%d\n", status.MPI\_SOURCE, sum, balance);

for (int i = 0; i < (p1 - p2) / 2; i += 2)

{

\*(p2 + i) = \*(p2 + i + 2);

\*(p2 + i + 1) = \*(p2 + i + 3);

}

p1 -= 2; //текущий указатель в конец очереди

} //end if

else //нельзя выдать, если нет у банка денег

{

pr = 0;

MPI\_Send(&pr, 1, MPI\_INT, m, msgtag, comm);

fprintf(fres, "банк не может выдать %d $ клиенту %d, т.к balance=%d\n", sum, m, balance);

p2 += 2; //конец очереди

}

}

kmax = (size - 1) - (p1 - nmag) / 2;

//fprintf(fres, "balance=%d; клиент %d сумма %d\n", balance, m, sum);

}

else //клиент

{

if (pr)

{

MPI\_Send(z, n, MPI\_INT, 0, msgtag, comm);

fprintf(fres, "клиент %d отправил запрос z[1]=%d на итерации l=%d\n", rank, z[1], l);

}

MPI\_Recv(&pr, 1, MPI\_INT, 0, msgtag, comm, &status);

if (pr)//=1

{

z[0] = rand() % 2;

z[1] = ((rand() % 100) \* rank) + 1;

fprintf(fres, "клиент %d; имеет запрос z[0]=%d и сумму z[1]=%d\n", rank, z[0], z[1]);

}

}

MPI\_Barrier(comm);

fclose(fres);

}

MPI\_Finalize();

return 0;

}

**Результаты:**

**mpiexec -n 3**

**result0.txt**

клиент 2 PUT = 135$, Текущий balance = 235

клиент 1 PUT = 68$, Текущий balance = 303

банк выдает клиенту 2 сумму z[1] = 1 Текущий balance = 302

банк выдает клиенту 1 сумму z[1] = 1 Текущий balance = 301

клиент 2 PUT = 49$, Текущий balance = 350

клиент 1 PUT = 25$, Текущий balance = 375

банк выдает клиенту 1 сумму z[1] = 59 Текущий balance = 316

банк выдает клиенту 2 сумму z[1] = 117 Текущий balance = 199

банк выдает клиенту 2 сумму z[1] = 129 Текущий balance = 70

в очереди: клиент 1 имеет сумму z[1] 65

банк не может выдать 65 $ клиенту 1, т.к balance=70

клиент 2 PUT = 91$, Текущий balance = 161

банк может выдать 2 клиенту сумму 65, текущий balance=96

клиент 2 PUT = 55$, Текущий balance = 151

клиент 1 PUT = 46$, Текущий balance = 197

клиент 1 PUT = 28$, Текущий balance = 225

клиент 2 PUT = 183$, Текущий balance = 408

клиент 2 PUT = 85$, Текущий balance = 493

клиент 1 PUT = 92$, Текущий balance = 585

клиент 2 PUT = 73$, Текущий balance = 658

клиент 1 PUT = 43$, Текущий balance = 701

клиент 1 PUT = 37$, Текущий balance = 738

клиент 2 PUT = 9$, Текущий balance = 747

банк выдает клиенту 2 сумму z[1] = 107 Текущий balance = 640

клиент 1 PUT = 5$, Текущий balance = 645

банк выдает клиенту 2 сумму z[1] = 165 Текущий balance = 480

банк выдает клиенту 1 сумму z[1] = 54 Текущий balance = 426

банк выдает клиенту 1 сумму z[1] = 83 Текущий balance = 343

клиент 2 PUT = 33$, Текущий balance = 376

клиент 1 PUT = 17$, Текущий balance = 393

банк выдает клиенту 2 сумму z[1] = 191 Текущий balance = 202

банк выдает клиенту 1 сумму z[1] = 96 Текущий balance = 106

клиент 2 PUT = 53$, Текущий balance = 159

клиент 1 PUT = 27$, Текущий balance = 186

клиент 2 PUT = 77$, Текущий balance = 263

клиент 1 PUT = 39$, Текущий balance = 302

клиент 2 PUT = 25$, Текущий balance = 327

клиент 1 PUT = 13$, Текущий balance = 340

клиент 2 PUT = 199$, Текущий balance = 539

клиент 1 PUT = 100$, Текущий balance = 639

клиент 2 PUT = 189$, Текущий balance = 828

**result1.txt**

клиент 1 имеет статус z[0] 1 и сумму z[1] 68

клиент 1 отправил запрос z[1]=68 на итерации l=0

клиент 1; имеет запрос z[0]=0 и сумму z[1]=1

клиент 1 отправил запрос z[1]=1 на итерации l=1

клиент 1; имеет запрос z[0]=1 и сумму z[1]=25

клиент 1 отправил запрос z[1]=25 на итерации l=2

клиент 1; имеет запрос z[0]=0 и сумму z[1]=59

клиент 1 отправил запрос z[1]=59 на итерации l=3

клиент 1; имеет запрос z[0]=0 и сумму z[1]=65

клиент 1 отправил запрос z[1]=65 на итерации l=4

клиент 1; имеет запрос z[0]=1 и сумму z[1]=46

клиент 1 отправил запрос z[1]=46 на итерации l=6

клиент 1; имеет запрос z[0]=1 и сумму z[1]=28

клиент 1 отправил запрос z[1]=28 на итерации l=7

клиент 1; имеет запрос z[0]=1 и сумму z[1]=92

клиент 1 отправил запрос z[1]=92 на итерации l=8

клиент 1; имеет запрос z[0]=1 и сумму z[1]=43

клиент 1 отправил запрос z[1]=43 на итерации l=9

клиент 1; имеет запрос z[0]=1 и сумму z[1]=37

клиент 1 отправил запрос z[1]=37 на итерации l=10

клиент 1; имеет запрос z[0]=1 и сумму z[1]=5

клиент 1 отправил запрос z[1]=5 на итерации l=11

клиент 1; имеет запрос z[0]=0 и сумму z[1]=54

клиент 1 отправил запрос z[1]=54 на итерации l=12

клиент 1; имеет запрос z[0]=0 и сумму z[1]=83

клиент 1 отправил запрос z[1]=83 на итерации l=13

клиент 1; имеет запрос z[0]=1 и сумму z[1]=17

клиент 1 отправил запрос z[1]=17 на итерации l=14

клиент 1; имеет запрос z[0]=0 и сумму z[1]=96

клиент 1 отправил запрос z[1]=96 на итерации l=15

клиент 1; имеет запрос z[0]=1 и сумму z[1]=27

клиент 1 отправил запрос z[1]=27 на итерации l=16

клиент 1; имеет запрос z[0]=1 и сумму z[1]=39

клиент 1 отправил запрос z[1]=39 на итерации l=17

клиент 1; имеет запрос z[0]=1 и сумму z[1]=13

клиент 1 отправил запрос z[1]=13 на итерации l=18

клиент 1; имеет запрос z[0]=1 и сумму z[1]=100

клиент 1 отправил запрос z[1]=100 на итерации l=19

клиент 1; имеет запрос z[0]=1 и сумму z[1]=95

**result2.txt**

клиент 2 имеет статус z[0] 1 и сумму z[1] 135

клиент 2 отправил запрос z[1]=135 на итерации l=0

клиент 2; имеет запрос z[0]=0 и сумму z[1]=1

клиент 2 отправил запрос z[1]=1 на итерации l=1

клиент 2; имеет запрос z[0]=1 и сумму z[1]=49

клиент 2 отправил запрос z[1]=49 на итерации l=2

клиент 2; имеет запрос z[0]=0 и сумму z[1]=117

клиент 2 отправил запрос z[1]=117 на итерации l=3

клиент 2; имеет запрос z[0]=0 и сумму z[1]=129

клиент 2 отправил запрос z[1]=129 на итерации l=4

клиент 2; имеет запрос z[0]=1 и сумму z[1]=91

клиент 2 отправил запрос z[1]=91 на итерации l=5

клиент 2; имеет запрос z[0]=1 и сумму z[1]=55

клиент 2 отправил запрос z[1]=55 на итерации l=6

клиент 2; имеет запрос z[0]=1 и сумму z[1]=183

клиент 2 отправил запрос z[1]=183 на итерации l=7

клиент 2; имеет запрос z[0]=1 и сумму z[1]=85

клиент 2 отправил запрос z[1]=85 на итерации l=8

клиент 2; имеет запрос z[0]=1 и сумму z[1]=73

клиент 2 отправил запрос z[1]=73 на итерации l=9

клиент 2; имеет запрос z[0]=1 и сумму z[1]=9

клиент 2 отправил запрос z[1]=9 на итерации l=10

клиент 2; имеет запрос z[0]=0 и сумму z[1]=107

клиент 2 отправил запрос z[1]=107 на итерации l=11

клиент 2; имеет запрос z[0]=0 и сумму z[1]=165

клиент 2 отправил запрос z[1]=165 на итерации l=12

клиент 2; имеет запрос z[0]=1 и сумму z[1]=33

клиент 2 отправил запрос z[1]=33 на итерации l=13

клиент 2; имеет запрос z[0]=0 и сумму z[1]=191

клиент 2 отправил запрос z[1]=191 на итерации l=14

клиент 2; имеет запрос z[0]=1 и сумму z[1]=53

клиент 2 отправил запрос z[1]=53 на итерации l=15

клиент 2; имеет запрос z[0]=1 и сумму z[1]=77

клиент 2 отправил запрос z[1]=77 на итерации l=16

клиент 2; имеет запрос z[0]=1 и сумму z[1]=25

клиент 2 отправил запрос z[1]=25 на итерации l=17

клиент 2; имеет запрос z[0]=1 и сумму z[1]=199

клиент 2 отправил запрос z[1]=199 на итерации l=18

клиент 2; имеет запрос z[0]=1 и сумму z[1]=189

клиент 2 отправил запрос z[1]=189 на итерации l=19

клиент 2; имеет запрос z[0]=1 и сумму z[1]=23

**Текст программы без стратегии:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <mpi.h>

#include <ctime>

#include <iostream>

#include <Windows.h>

using std::cout;

using std::cin;

using std::endl;

const int n = 2,

m = 5,

lmax = 15;//жизненный цикл

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

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

int rank, size;// rank -номер процесса, size - количество процессов

MPI\_Status status;

if (MPI\_Init(&argc, &argv) != MPI\_SUCCESS)

return 1;

if (MPI\_Comm\_size(MPI\_COMM\_WORLD, &size) != MPI\_SUCCESS)

{

MPI\_Finalize();

return 2;

}

if (MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank) != MPI\_SUCCESS)

{

MPI\_Finalize();

return 3;

}

int msgtag = 7,

pr = 1,

mag[m \* n],

balance,

z[n],

\* p1,

\* nmag,

M,

kmax,

sum,

N = 5;

p1 = nmag = mag;

srand((unsigned)time(0));

FILE\* f;

char name[20];

sprintf(name, "process%d.txt", rank);

fopen\_s(&f, name, "w");

if (!rank) {

//процесс-банк

balance = 200;

kmax = size - 1;

}

else {

//процессы-клиенты

z[0] = rand() % 2;

z[1] = ((rand() % 200) \* rank) + 1;

fprintf(f, "Процесс %d имеет статус z[0] %d и средства z[1] %d\n", rank, z[0], z[1]);

}

fclose(f);

MPI\_Barrier(MPI\_COMM\_WORLD);

for (int l = 0; l < lmax; l++) //ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММЫ

{

fopen\_s(&f, name, "a");

if (!rank) //банк

{

for (int i = 0; i < kmax; i++) //Работаем с клиентами по очереди

{

MPI\_Recv(z, n, MPI\_INT, MPI\_ANY\_SOURCE, msgtag, MPI\_COMM\_WORLD, &status);

if (z[0]) //пополнение СЧЁТА

{

balance += z[1];

pr = 1;

MPI\_Send(&pr, 1, MPI\_INT, status.MPI\_SOURCE, msgtag, MPI\_COMM\_WORLD);

fprintf(f, "Процесс-клиент %d положил сумму = %d$, Текущий СЧЁТ = %d\n", status.MPI\_SOURCE, z[1], balance);

}

else //снятие со СЧЁТА

{

if (z[1] > balance - N) //помещаем в очередь

{

fprintf(f, "В очереди: процесс-клиент %d хочет снять сумму z[1] %d\n", status.MPI\_SOURCE, z[1]);

\*p1 = status.MPI\_SOURCE; //указатель на клиента

p1++;

\*p1 = z[1];

p1++;

}

else

{

balance -= z[1];

pr = 1;

MPI\_Send(&pr, 1, MPI\_INT, status.MPI\_SOURCE, msgtag, MPI\_COMM\_WORLD);

fprintf(f, "Банк выдает клиенту %d сумму z[1] = %d Текущий СЧЁТ = %d\n", status.MPI\_SOURCE, z[1], balance);

}

}

}

//РАБОТА С ОЧЕРЕДЬЮ идёт в обратном порядке: начинаем с последнего пришедшего клиента

int flag = 1;

while ((p1 != nmag) && flag)//пока очередь не пустая

{

fprintf(f, "l=%d очередь не пуста, количество клиентов равно %d\n", l, (p1 - nmag) / 2);

M = \*(p1 - 2); //Какому процессу выдаём

sum = \*(p1 - 1);

if (sum < balance - N) //Если сумма которую хотят снять меньше Текущего СЧЁТА, то можно выдать деньги

{

balance -= sum;

pr = 1;

MPI\_Send(&pr, 1, MPI\_INT, M, msgtag, MPI\_COMM\_WORLD);

fprintf(f, "l = % d банк выдал процессу % d сумму - % d и клиент выбыл из очереди\n", l, M, sum);

p1 -= 2; //указатель сдвигаем в конец очереди

}

else //Сумма, которую хотят взять больше, чем остаток средств на счёте, следовательно банк не может выдать средства

{

flag = 0;

fprintf(f, "В банке balance = %d, а процесс %d просит %d\n", balance, M, sum);

}

}

//Проверим оставшуюся очередь. Если все процессы-клиенты стоят в очереди, закрываем банк

kmax = (p1 - nmag) / 2;

for (int i = 1; i <= kmax; i++) {

pr = 0;

M = \*(p1 - 2 \* i);

MPI\_Send(&pr, 1, MPI\_INT, M, msgtag, MPI\_COMM\_WORLD);

fprintf(f, "l=%d банк не выдал процессу %d сумму - %d и клиент остался в очереди\n", l, M, \*(p1 - 2 \* i + 1));

}

kmax = (size - 1) - kmax;

fprintf(f, "kmax = %d\n", kmax);

if (!kmax) fprintf(f, "В банке нет достаточно денег \n");

}

else //клиент

{

if (pr)

{

MPI\_Send(z, n, MPI\_INT, 0, msgtag, MPI\_COMM\_WORLD);

fprintf(f, "Процесс-клиент %d отправил запрос z[1]=%d на итерации l=%d\n", rank, z[1], l);

}

MPI\_Recv(&pr, 1, MPI\_INT, 0, msgtag, MPI\_COMM\_WORLD, &status);

if (pr)//=1

{

z[0] = rand() % 2;

z[1] = ((rand() % 100) \* 1) + 1;

fprintf(f, "Процесс-клиент %d имеет запрос z[0]=%d и сумму z[1]=%d\n", rank, z[0], z[1]);

}

}

MPI\_Barrier(MPI\_COMM\_WORLD);

fclose(f);

}

MPI\_Finalize();

return 0;

}

**Результаты:**

**mpiexec -n 3**

**process0.txt**

Процесс-клиент 1 положил сумму = 38$, Текущий СЧЁТ = 238

Процесс-клиент 2 положил сумму = 75$, Текущий СЧЁТ = 313

kmax = 2

Процесс-клиент 1 положил сумму = 25$, Текущий СЧЁТ = 338

Процесс-клиент 2 положил сумму = 25$, Текущий СЧЁТ = 363

kmax = 2

Банк выдает клиенту 2 сумму z[1] = 54 Текущий СЧЁТ = 309

Банк выдает клиенту 1 сумму z[1] = 54 Текущий СЧЁТ = 255

kmax = 2

Процесс-клиент 1 положил сумму = 86$, Текущий СЧЁТ = 341

Процесс-клиент 2 положил сумму = 86$, Текущий СЧЁТ = 427

kmax = 2

Процесс-клиент 1 положил сумму = 93$, Текущий СЧЁТ = 520

Процесс-клиент 2 положил сумму = 93$, Текущий СЧЁТ = 613

kmax = 2

Процесс-клиент 2 положил сумму = 4$, Текущий СЧЁТ = 617

Процесс-клиент 1 положил сумму = 4$, Текущий СЧЁТ = 621

kmax = 2

Банк выдает клиенту 1 сумму z[1] = 47 Текущий СЧЁТ = 574

Банк выдает клиенту 2 сумму z[1] = 47 Текущий СЧЁТ = 527

kmax = 2

Процесс-клиент 1 положил сумму = 8$, Текущий СЧЁТ = 535

Процесс-клиент 2 положил сумму = 8$, Текущий СЧЁТ = 543

kmax = 2

Банк выдает клиенту 2 сумму z[1] = 15 Текущий СЧЁТ = 528

Банк выдает клиенту 1 сумму z[1] = 15 Текущий СЧЁТ = 513

kmax = 2

Банк выдает клиенту 1 сумму z[1] = 38 Текущий СЧЁТ = 475

Банк выдает клиенту 2 сумму z[1] = 38 Текущий СЧЁТ = 437

kmax = 2

Банк выдает клиенту 1 сумму z[1] = 80 Текущий СЧЁТ = 357

Банк выдает клиенту 2 сумму z[1] = 80 Текущий СЧЁТ = 277

kmax = 2

Процесс-клиент 1 положил сумму = 15$, Текущий СЧЁТ = 292

Процесс-клиент 2 положил сумму = 15$, Текущий СЧЁТ = 307

kmax = 2

Банк выдает клиенту 1 сумму z[1] = 19 Текущий СЧЁТ = 288

Банк выдает клиенту 2 сумму z[1] = 19 Текущий СЧЁТ = 269

kmax = 2

Банк выдает клиенту 2 сумму z[1] = 50 Текущий СЧЁТ = 219

Банк выдает клиенту 1 сумму z[1] = 50 Текущий СЧЁТ = 169

kmax = 2

Банк выдает клиенту 1 сумму z[1] = 34 Текущий СЧЁТ = 135

Банк выдает клиенту 2 сумму z[1] = 34 Текущий СЧЁТ = 101

kmax = 2

**process1.txt**

Процесс 1 имеет статус z[0] 1 и средства z[1] 38

Процесс-клиент 1 отправил запрос z[1]=38 на итерации l=0

Процесс-клиент 1 имеет запрос z[0]=1 и сумму z[1]=25

Процесс-клиент 1 отправил запрос z[1]=25 на итерации l=1

Процесс-клиент 1 имеет запрос z[0]=0 и сумму z[1]=54

Процесс-клиент 1 отправил запрос z[1]=54 на итерации l=2

Процесс-клиент 1 имеет запрос z[0]=1 и сумму z[1]=86

Процесс-клиент 1 отправил запрос z[1]=86 на итерации l=3

Процесс-клиент 1 имеет запрос z[0]=1 и сумму z[1]=93

Процесс-клиент 1 отправил запрос z[1]=93 на итерации l=4

Процесс-клиент 1 имеет запрос z[0]=1 и сумму z[1]=4

Процесс-клиент 1 отправил запрос z[1]=4 на итерации l=5

Процесс-клиент 1 имеет запрос z[0]=0 и сумму z[1]=47

Процесс-клиент 1 отправил запрос z[1]=47 на итерации l=6

Процесс-клиент 1 имеет запрос z[0]=1 и сумму z[1]=8

Процесс-клиент 1 отправил запрос z[1]=8 на итерации l=7

Процесс-клиент 1 имеет запрос z[0]=0 и сумму z[1]=15

Процесс-клиент 1 отправил запрос z[1]=15 на итерации l=8

Процесс-клиент 1 имеет запрос z[0]=0 и сумму z[1]=38

Процесс-клиент 1 отправил запрос z[1]=38 на итерации l=9

Процесс-клиент 1 имеет запрос z[0]=0 и сумму z[1]=80

Процесс-клиент 1 отправил запрос z[1]=80 на итерации l=10

Процесс-клиент 1 имеет запрос z[0]=1 и сумму z[1]=15

Процесс-клиент 1 отправил запрос z[1]=15 на итерации l=11

Процесс-клиент 1 имеет запрос z[0]=0 и сумму z[1]=19

Процесс-клиент 1 отправил запрос z[1]=19 на итерации l=12

Процесс-клиент 1 имеет запрос z[0]=0 и сумму z[1]=50

Процесс-клиент 1 отправил запрос z[1]=50 на итерации l=13

Процесс-клиент 1 имеет запрос z[0]=0 и сумму z[1]=34

Процесс-клиент 1 отправил запрос z[1]=34 на итерации l=14

Процесс-клиент 1 имеет запрос z[0]=1 и сумму z[1]=53

**process2.txt**

Процесс 2 имеет статус z[0] 1 и средства z[1] 75

Процесс-клиент 2 отправил запрос z[1]=75 на итерации l=0

Процесс-клиент 2 имеет запрос z[0]=1 и сумму z[1]=25

Процесс-клиент 2 отправил запрос z[1]=25 на итерации l=1

Процесс-клиент 2 имеет запрос z[0]=0 и сумму z[1]=54

Процесс-клиент 2 отправил запрос z[1]=54 на итерации l=2

Процесс-клиент 2 имеет запрос z[0]=1 и сумму z[1]=86

Процесс-клиент 2 отправил запрос z[1]=86 на итерации l=3

Процесс-клиент 2 имеет запрос z[0]=1 и сумму z[1]=93

Процесс-клиент 2 отправил запрос z[1]=93 на итерации l=4

Процесс-клиент 2 имеет запрос z[0]=1 и сумму z[1]=4

Процесс-клиент 2 отправил запрос z[1]=4 на итерации l=5

Процесс-клиент 2 имеет запрос z[0]=0 и сумму z[1]=47

Процесс-клиент 2 отправил запрос z[1]=47 на итерации l=6

Процесс-клиент 2 имеет запрос z[0]=1 и сумму z[1]=8

Процесс-клиент 2 отправил запрос z[1]=8 на итерации l=7

Процесс-клиент 2 имеет запрос z[0]=0 и сумму z[1]=15

Процесс-клиент 2 отправил запрос z[1]=15 на итерации l=8

Процесс-клиент 2 имеет запрос z[0]=0 и сумму z[1]=38

Процесс-клиент 2 отправил запрос z[1]=38 на итерации l=9

Процесс-клиент 2 имеет запрос z[0]=0 и сумму z[1]=80

Процесс-клиент 2 отправил запрос z[1]=80 на итерации l=10

Процесс-клиент 2 имеет запрос z[0]=1 и сумму z[1]=15

Процесс-клиент 2 отправил запрос z[1]=15 на итерации l=11

Процесс-клиент 2 имеет запрос z[0]=0 и сумму z[1]=19

Процесс-клиент 2 отправил запрос z[1]=19 на итерации l=12

Процесс-клиент 2 имеет запрос z[0]=0 и сумму z[1]=50

Процесс-клиент 2 отправил запрос z[1]=50 на итерации l=13

Процесс-клиент 2 имеет запрос z[0]=0 и сумму z[1]=34

Процесс-клиент 2 отправил запрос z[1]=34 на итерации l=14

Процесс-клиент 2 имеет запрос z[0]=1 и сумму z[1]=53