МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

По лабораторной работе № \_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

Проверил\_\_:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Брест – 2023

Ход работы

*Код программы*

void ClientRecvHandler() {

while (true) {

switch (state) {

case getting\_Men: {

g\_lock.lock();

recv(Connection, msg, sizeof(msg), NULL);

Men = msg;

state = getting\_Ben;

g\_lock.unlock();

break;

}

case getting\_Ben: {

g\_lock.lock();

recv(Connection, msg, sizeof(msg), NULL);

ksecr = msg;

state = getting\_ksecr;

g\_lock.unlock();

break;

}

case getting\_ksecr: {

g\_lock.lock();

recv(Connection, msg, sizeof(msg), NULL);

Ben = msg;

state = sending\_results;

g\_lock.unlock();

break;

}

default: break;

}

if (state == close) {

break;

}

}

}

void ClientSendHandler() {

while (true) {

switch (state) {

case sending\_M: {

g\_lock.lock();

string M = generate\_M();

in\_msg(M);

send(Connection, msg, sizeof(msg), NULL);

state = getting\_Men;

g\_lock.unlock();

break;

}

case sending\_results: {

g\_lock.lock();

string B = rc4.crypt(Ben, ksecr);

in\_msg(B);

send(Connection, msg, sizeof(msg), NULL);

state = close;

g\_lock.unlock();

break;

}

default: break;

}

if (state == close) {

cout << "Men: " << Men << endl;

cout << "Ben: " << Ben << endl;

cout << "ksecr: " << ksecr << endl;

break;

}

}

}

void ClientRecvHandler() {

while (true) {

switch (state) {

case getting\_M: {

g\_lock.lock();

recv(Connection, msg, sizeof(msg), MSG\_WAITALL);

M = msg;

state = sending\_Men;

g\_lock.unlock();

break;

}

case getting\_results: {

g\_lock.lock();

recv(Connection, msg, sizeof(msg), MSG\_WAITALL);

results = msg;

state = close;

g\_lock.unlock();

break;

}

default: break;

}

if (state == close) {

cout << "M: " << M << endl;

cout << "ksecr: " << ksecr << endl;

cout << "B: " << B << endl;

cout << "results: " << results << endl;

break;

}

}

}

void ClientSendHandler() {

while (true) {

switch (state) {

case sending\_Men: {

g\_lock.lock();

string Men = rc4.crypt(M, ksecr);

in\_msg(Men);

send(Connection, msg, sizeof(msg), NULL);

state = sending\_Ben;

g\_lock.unlock();

break;

}

case sending\_Ben: {

g\_lock.lock();

in\_msg(ksecr);

send(Connection, msg, sizeof(msg), NULL);

state = sending\_ksecr;

g\_lock.unlock();

break;

}

case sending\_ksecr: {

g\_lock.lock();

string Ben = rc4.crypt(B, ksecr);

in\_msg(Ben);

send(Connection, msg, sizeof(msg), NULL);

state = getting\_results;

g\_lock.unlock();

break;

}

default: break;

}

if (state == close) {

break;

}

}

}

*Вывод программы*

