МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра информационных систем

Отчет   
по лабораторной работе №13  
на тему: «Реализация клиент-серверных приложений»  
по дисциплине «Операционные системы»

Выполнили: Марочкин М.А. Шифр: 170584  
 Яшин М. О. Шифр: 170133  
 Шорин В.Д. Шифр: 171406  
ИПАИТ  
Направление: 09.03.04 «Программная инженерия»  
Группа: 71-ПГ  
Проверил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Отметка о зачете:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата «\_\_\_\_» \_\_\_\_\_\_ 2018 г.

Орел, 2018 г.

**Ход работы**

1. Разработать клиент-серверное приложение реализующее обмен сообщениями, зашифрованными с помощью алгоритма RSA, между клиентов и сервером.

Программный код приложения-сервера:

#include <stdio.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <iostream>

#include <arpa/inet.h>

#include <unistd.h>

#include <stdlib.h>

#include <vector>

using namespace std;

long long int NOD(unsigned long int n1, long long int n2){

long long int div;

if (n1 == n2)

return n1;

long long int d = n1 - n2;

if (d < 0){

d = -d;

div = NOD(n1, d);

}

else{

div = NOD(n2, d);

}

return div;

}

long long int myPow(int smb, int e, int n){

long long int res;

smb %= n;

res = smb;

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

res = (res \* smb) % n;

}

return res;

}

int main(){

srand(time(NULL));

int simpleNumbers[] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 73, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199};

int p = simpleNumbers[rand() % (45 + 1)];

int q = simpleNumbers[rand() % (45 + 1)];

long long int n;

long long int e = 0;

long long int d;

long long int pqminus;

unsigned long long int ed;

unsigned long long int tmp;

vector<unsigned long long int> message;

int maximum = 1000000;

int minimum = 1;

d = minimum + rand() % ((maximum + 1) - minimum);

n = p\*q;

pqminus = (q - 1)\*(p - 1);

while (NOD(d, pqminus) != 1){

d++;

}

e = minimum + rand() % ((maximum+ 1) - minimum);

ed = e \* d;

while ((ed) % pqminus != 1){

e++;

ed = e \* d;

}

int sock, listener;

struct sockaddr\_in addr;

int cnt;

int id;

listener = socket(AF\_INET, SOCK\_STREAM, 0);

if(listener < 0){

perror("socket");

exit(1);

}

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(3425);

addr.sin\_addr.s\_addr = INADDR\_ANY;

if(bind(listener, (struct sockaddr \*)&addr, sizeof(addr)) < 0){

perror("bind");

exit(2);

}

listen(listener, 1);

while(1){

sock = accept(listener, NULL, NULL);

if(sock < 0){

perror("accept");

exit(3);

}

switch(fork()){

case -1:

perror("fork");

break;

case 0:

close(listener);

read(sock, &id, sizeof(int));

cout << "User's ID: " << id << " ----> Connected" << endl << endl;

write(sock, &e, sizeof(long long int));

write(sock, &n, sizeof(long long int));

while(1){

read(sock, &cnt, sizeof(int));

if (cnt > 0){

for(int i = 0; i < cnt; i++){

read(sock, &tmp, sizeof(unsigned long long int));

message.push\_back(tmp);

}

cout << "User's ID: " << id << endl;

cout << "Secret message: ";

for (int i = 0; i < cnt; i++){

cout << message[i];

}

cout << endl;

cout << "OutputMessage: ";

for (int i = 0; i < cnt; i++){

cout << (char)myPow(message[i], d, n);

}

cout << endl << endl;

message.clear();

cnt = 0;

}

else {

close(sock);

cout << "User's ID: " << id << " ----> Disconnected" << endl << endl;

\_exit(0);

}

}

default:

close(sock);

}

}

close(listener);

return 0;

}

Программный код приложения-клиента:

#include <stdio.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <iostream>

#include <arpa/inet.h>

#include <unistd.h>

#include <stdlib.h>

#include <vector>

using namespace std;

long long int myPow(int smb, int e, int n){

long long int res;

smb %= n;

res = smb;

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

res = (res \* smb) % n;

}

return res;

}

int main(){

long long int e, n;

string msg;

vector<unsigned long long int> message;

int sock;

int cnt = 4;

int id;

struct sockaddr\_in addr;

sock = socket(AF\_INET, SOCK\_STREAM, 0);

if(sock < 0){

perror("socket");

exit(1);

}

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(3425);

addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

if(connect(sock, (struct sockaddr \*)&addr, sizeof(addr)) < 0){

perror("connect");

exit(2);

}

cout << "Enter ID: ";

cin >> id;

write(sock, &id, sizeof(int));

read(sock, &e, sizeof(long long int));

read(sock, &n, sizeof(long long int));

cout << "Connected..." << endl;

getchar();

while (true){

cout << endl << "Write message: ";

//getchar();

getline(cin, msg);

cout << "Secret message: ";

for (int i = 0; i < msg.size(); i++){

message.push\_back(myPow((int)msg[i], e, n));

cout << message[i];

}

cout << endl;

cnt = message.size();

write(sock, &cnt, sizeof(int));

for(int i = 0; i < cnt; i++){

write(sock, &message[i], sizeof(unsigned long long int));

}

message.clear();

}

close(sock);

return 0;

}

2. Разработать клиент-серверное приложение реализующее обмен сообщениями между клиентами через сервер.

Программный код приложения-сервера:

#include <sys/types.h>

#include <sys/socket.h>

#include <sys/time.h>

#include <netinet/in.h>

#include <stdio.h>

#include <unistd.h>

#include <fcntl.h>

#include <algorithm>

#include <set>

#include <vector>

#include <string.h>

#include <iostream>

using namespace std;

int main(){

int listener;

struct sockaddr\_in addr;

char buf[256];

int bytes\_read;

vector <bool> flags;

char del[] = "";

int sendID, curSockID = 0;

listener = socket(AF\_INET, SOCK\_STREAM, 0);

if(listener < 0){

perror("socket");

exit(1);

}

fcntl(listener, F\_SETFL, O\_NONBLOCK);

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(3425);

addr.sin\_addr.s\_addr = INADDR\_ANY;

if(bind(listener, (struct sockaddr \*)&addr, sizeof(addr)) < 0){

perror("bind");

exit(2);

}

listen(listener, 5);

set<int> clients;

clients.clear();

while(1){

fd\_set readset;

FD\_ZERO(&readset);

FD\_SET(listener, &readset);

for(set<int>::iterator it = clients.begin(); it != clients.end(); it++){

FD\_SET(\*it, &readset);

}

timeval timeout;

timeout.tv\_sec = 300;

timeout.tv\_usec = 0;

int mx = max(listener, \*max\_element(clients.begin(), clients.end()));

if(select(mx+1, &readset, NULL, NULL, &timeout) <= 0){

perror("select");

exit(3);

}

if(FD\_ISSET(listener, &readset)){

int sock = accept(listener, NULL, NULL);

if(sock < 0){

perror("accept");

exit(3);

}

write(sock, &curSockID, sizeof(int));

cout << "Client " << curSockID << " ----> Connected" << endl << endl;

curSockID++;

flags.push\_back(true);

clients.insert(sock);

}

int i;

set<int>::iterator it;

for(i = 0, it = clients.begin(); it != clients.end(); it++, i++){

if(FD\_ISSET(\*it, &readset)){

if(flags[i] == true){

strcpy(buf, del);

bytes\_read = recv(\*it, &sendID, sizeof(int), 0);

if(bytes\_read <= 0){

flags[i] = false;

cout << "Client " << i << " ----> Disconnected" << endl << endl;

continue;

}

if (sendID < flags.size()){

cout << "SEND ID: " << sendID << endl;

if (recv(\*it, buf, 256, 0) > 0){

cout << "Message: ";

for(int i = 0; i < strlen(buf); i++){

buf[i] ^= i % strlen(buf);

}

buf[strlen(buf)] = '\0';

printf("%s", buf);

cout << endl;

set<int>::iterator sendIt;

sendIt = clients.begin();

advance(sendIt, sendID);

send(\*sendIt, buf, (strlen(buf)+1)\*sizeof(char), 0);

}

}

else {

cout << "ID " << sendID << " incorrect!" << endl << endl;

}

}

}

}

}

return 0;

}

Программный код приложения-клиента:

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <iostream>

#include <arpa/inet.h>

#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

#include <vector>

#include <string.h>

#include <thread>

using namespace std;

char buf[256];

char del[] = "";

int sock;

int sendID, curSockID;

struct sockaddr\_in addr;

void ListenThread(){

while(1){

if (recvfrom(sock, buf, 256, 0, NULL, NULL) > 0){

cout << endl << endl << "Input message: ";

printf("%s", buf);

cout << endl;

}

}

}

void SendThread(){

while(1){

cout << endl << "Write id to send and message: ";

cin >> sendID;

send(sock, &sendID, sizeof(int), 0);

getchar();

fgets(buf, 256, stdin);

for(int i = 0; i < strlen(buf); i++){

buf[i] ^= i % strlen(buf);

}

send(sock, buf, (strlen(buf)+1)\*sizeof(char), 0);

buf[strlen(buf)] = '\0';

}

}

int main(){

sock = socket(AF\_INET, SOCK\_STREAM, 0);

if(sock < 0){

perror("socket");

exit(1);

}

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(3425);

addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

if(connect(sock, (struct sockaddr \*)&addr, sizeof(addr)) < 0){

perror("connect");

exit(2);

}

read(sock, &curSockID, sizeof(int));

cout << "Your ID: " << curSockID << endl;

thread ListThr(ListenThread);

ListThr.detach();

thread SendThr(SendThread);

SendThr.join();

cout << "End" << endl;

close(sock);

return 0;

}