## МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

## «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**Кафедра Вычислительной техники**

## ОТЧЕТ

**по лабораторной работе №11**

## по дисциплине «Организация процессов и программирование в среде Linux»

**Тема: Взаимодействие процессов через сокеты**

|  |  |
| --- | --- |
| Студент гр. 8308 | Петров Г.А. |
| Преподаватель | Разумовский Г.В. |

Санкт-Петербург 2021

# Цель работы

Целью лабораторной работы является знакомство с организацией сокетов и системными функциями, обеспечивающими работу с сокетами.

# Задание

Написать две программы (сервер и клиент), которые обмениваются сообщениями через потоковые сокеты. Клиенты проверяют возможность соединения с сервером и в случае отсутствия соединения или истечения времени ожидания отправки сообщения завершают работу. После соединения с сервером они генерируют случайную последовательность чисел и выводят ее на экран, а затем отсылают серверу. Сервер в течение определенного времени ждет запросы от клиентов и в случае их отсутствия завершает работу. При поступлении запроса от клиента сервер порождает обслуживающий процесс, который принимает последовательность чисел, упорядочивает ее и выводит на экран, а затем отсылает обратно клиенту и завершают работу. Клиент полученную последовательность выводит на экран и заканчивает свою работу.

# Примеры выполнения программы

Программы были разработаны и откомпилированы. После чего программы были запушена через два терминала. Результаты работы программ приведены на рисунках 1-3.



Рисунок 1

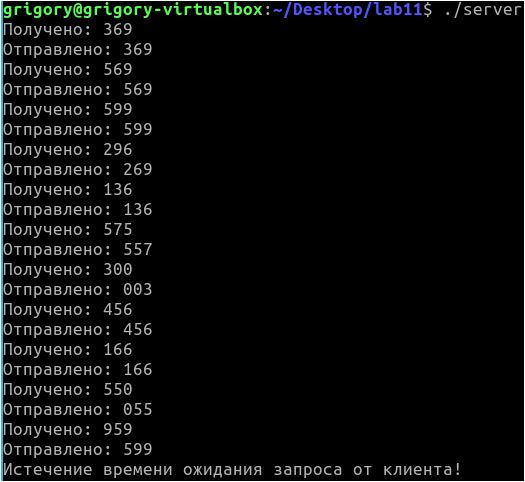


Рисунок 2

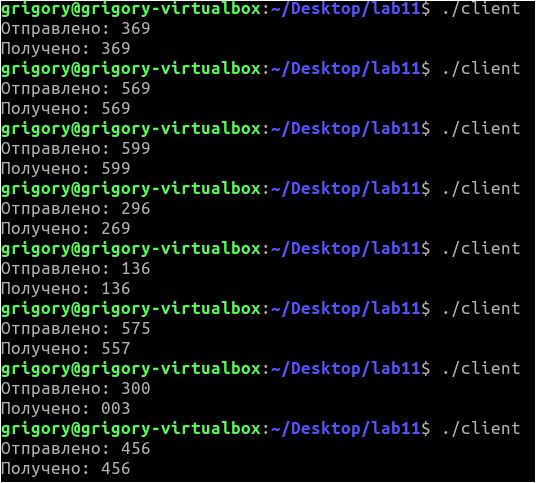


Рисунок 3

# Исходный код программ

## client.cpp

#include <iostream>

#include <sys/socket.h>

#include <netinet/in.h>

#include <unistd.h>

int main()

{

srand(time(0));

int SocID;

struct sockaddr\_in addr;

struct timeval tv;

fd\_set writefds;

int wv;

char message[3],buf[3];

//создание сокета

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

if(SocID<0)

{

perror("Error in socket()");

exit(1);

}

addr.sin\_family=AF\_INET;

addr.sin\_port=htons(7777);

addr.sin\_addr.s\_addr=htonl(INADDR\_LOOPBACK);

//подключение к серверу

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

{

perror("Error in connect()");

exit(2);

}

else

{//ожидание отправки сообщения

FD\_ZERO(&writefds);

FD\_SET(SocID,&writefds);

tv.tv\_sec=10;

tv.tv\_usec=0;

wv=select(FD\_SETSIZE,NULL,&writefds,NULL,&tv);

if(wv==0)

{

std::cout<<"Истечение времени ожидания отправки сообщения!"<<std::endl;

}

else

{

//отправка

std::sprintf(message,"%d",rand()%899+100);

send(SocID,message,sizeof(message),0);

std::cout<<"Отправлено: "<<message<<std::endl;

//получение

recv(SocID,buf,sizeof(message),0);

std::cout<<"Получено: "<<buf<<std::endl;

}

}

close(SocID);

return 0;

}

## server.cpp

#include <iostream>

#include <sys/socket.h>

#include <netinet/in.h>

#include <unistd.h>

int main()

{

int SocID, ServerID;

struct sockaddr\_in addr;

struct timeval tv;

fd\_set fds;

int cv;

pid\_t pid;

char SocID\_char[255];

//создание сокета

ServerID=socket(AF\_INET,SOCK\_STREAM,0);//создание принимающего сокета

if(ServerID<0)

{

perror("Error in socket()");

exit(1);

}

addr.sin\_family=AF\_INET;

addr.sin\_port=htons(7777);

addr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

//привязка к сетевому адресу

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

{

perror("Error in bind()");

exit(2);

}

listen(ServerID,1);

while(true)

{

//ожидание клиента

FD\_ZERO(&fds);

FD\_SET(ServerID,&fds);

tv.tv\_sec=5;

tv.tv\_usec=0;

cv=select(FD\_SETSIZE,&fds,NULL,NULL,&tv);

if(cv==0)

{

std::cout<<"Истечение времени ожидания запроса от клиента!"<<std::endl;

break;

}

else

{

SocID=accept(ServerID,NULL,NULL);

if(SocID<0)

{

perror("Error in accpt()");

exit(3);

}

//получение сообщения

FD\_ZERO(&fds);

FD\_SET(SocID,&fds);

tv.tv\_sec=5;

tv.tv\_usec=0;

cv=select(FD\_SETSIZE,&fds,NULL,NULL,&tv);

if(cv==0)

{

std::cout<<"Истечение времени ожидания сообщения от клиента!"<<std::endl;

close(SocID);

break;

}

else

{

pid=fork();

std::sprintf(SocID\_char,"%d",SocID);

if(pid==0)

execl("process"," ",SocID\_char,NULL);

close(SocID);

}

}

}

close(ServerID);

return 0;

}

## process.cpp

#include <iostream>

#include <sys/socket.h>

#include <netinet/in.h>

#include <unistd.h>

#include <algorithm>

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

{

char bufc[3];

int size\_buf;

int SocID=atoi(argv[1]);//сокет

size\_buf=recv(SocID,bufc,3,0);//получени

std::cout<<"Получено: "<<bufc<<std::endl;

std::sort(std::begin(bufc),std::end(bufc));//сортировка

send(SocID,bufc,size\_buf,0);//передача

std::cout<<"Отправлено: "<<bufc<<std::endl;

exit(0);

}

# Вывод

При выполнении лабораторной работы изучены и использованы сокеты и системные функции, обеспечивающие работу с сокетами. Программа разработанная в соответствии с заданием, работает корректно.