**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Томский государственный университет систем управления и радиоэлектроники»**

Кафедра компьютерных систем в управлении и проектировании (КСУП)

Отчет по лабораторной работе № 10 по курсу

«Операционные системы»

**Сетевые датаграммные каналы**

**Выполнил:**

студент гр. 589-3

\_\_\_\_\_\_\_\_\_\_\_\_\_Потлог А.M.

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

**Принял:**

к.т.н., доцент

\_\_\_\_\_\_\_\_Коцубинский В.П.

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

Томск 2020

**Цель работы:**

Целью выполнения настоящей лабораторной работы является получение навыков создания на основе сокетов датаграммных UDP-каналов, используемых для связи между удаленными процессами.

**Ход выполнения работы:**

1. В качестве IP адреса будем использовать константу INADDR\_ANY, которая содержит IP “своего” хоста.(Поле sin\_addr, структуры sockaddr\_in)
2. Пишем код server.c и client.c и компилируем(порт 1305).

<server.c>

#include <sys/socket.h>

#include <sys/types.h>

#include <stdlib.h>

#include <stdio.h>

#include <netinet/in.h>

#include <time.h>

#include <string.h>

#define LBUF 128

#define LEN sizeof(struct sockaddr\_in)

int main(){

struct sockaddr\_in addr, client;

char buffer[LBUF];

int n, sock, s\_len;

int c\_len = LEN;

char buffstr[50];

time\_t rawtime;

struct tm \*timeinfo;

char timebuffer[10];

FILE \*fil;

if ((sock = socket(AF\_INET, SOCK\_DGRAM , 0)) < 0){

printf("Server: Unable to create socket.\n");

exit(1);

}

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(1305);

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

s\_len = sizeof(addr);

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

printf("Server:Unable to bind socket.\n");

exit(1);

}

for(;;){

n=recvfrom(sock, buffer, LBUF, 0, (struct sockaddr\*) &client, &c\_len);

if (n<0)

{

printf("Server:Recieving error.\n");

continue;

}

time (&rawtime);

timeinfo = localtime (&rawtime);

strftime(timebuffer, 80, "%H:%M:%S: " , timeinfo);

buffer[n]='\0';

strcat(strcpy(buffstr, timebuffer), buffer);

printf("\n");

printf("%s",buffstr);

buffstr[n]='\n';

if ((fil = fopen("data","a")) == NULL){

printf("Server:Cannot create file data");

exit(1);

}

fputs(buffstr, fil);

fclose(fil);

char msg[] = "Potlog 589-3. you use my server";

sendto(sock, msg, LBUF , 0, (struct sockaddr \*) &client, c\_len);

}

}

<client.c>

#include <sys/types.h>

#include <sys/socket.h>

#include <stdio.h>

#include <stdlib.h>

#include <netinet/in.h>

#include <string.h>

#define LBUF 128

#define LEN sizeof(struct sockaddr\_in)

char buffer[LBUF];

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

if (argc!=2){

printf("Client: Need a file name to work.\n");

exit(1);

}

FILE \*fil;

if((fil=fopen(argv[1],"r")) == NULL){

printf("Client: Cannot open file.\n");

exit(1);

}

fread(buffer, sizeof(buffer),1,fil);

if(strlen(buffer)>40){

printf("Client: Datagram bigger than 40 bytes.\n");

exit(1);

}

struct sockaddr\_in addr;

int n, sock;

if((sock = socket(AF\_INET, SOCK\_DGRAM,0))<0){

printf("Client: Unable to create socket.\n");

exit(1);

}

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(1305);

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

if (sendto(sock,buffer,LBUF,0,(struct sockaddr\*)&addr, sizeof(addr)) != LBUF){

printf("Client: Unable to send.\n");

exit(1);

}

for(;;){

n=recvfrom(sock,buffer,LBUF,0,NULL,NULL);

if(n<0){

printf("Client: Recieving error.\n");continue;

}

buffer[n]='\0';

printf("Otvet servera: %s\n", buffer);

break;

}

fclose(fil);

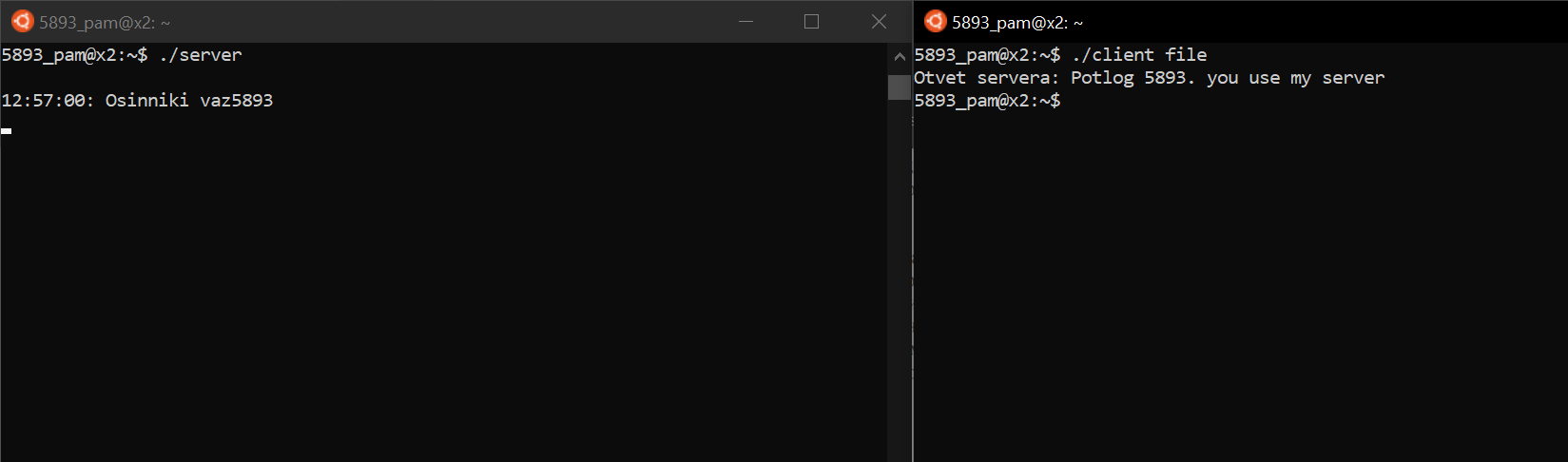
close(sock);

exit(0);

}

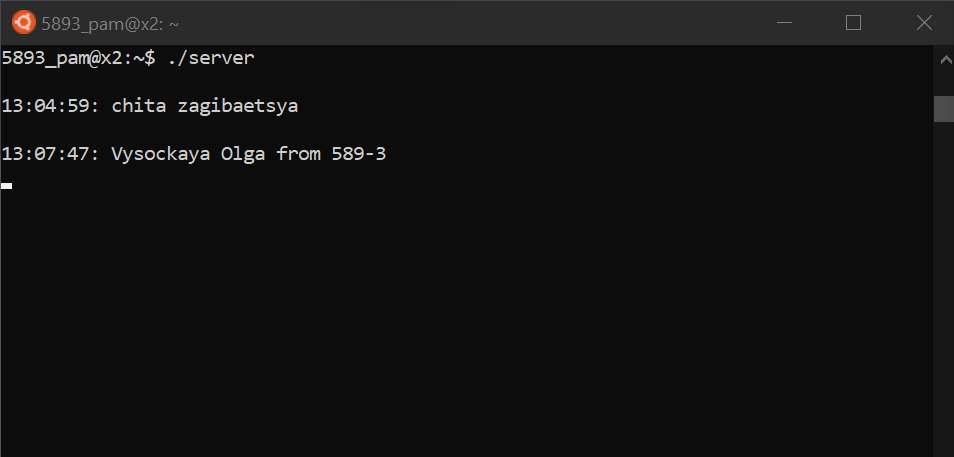
1. Совместная отладка своих клиентской и серверной частей в локальном режиме. В процессе данной отладки клиентская и серверная части располагаются на одном и том же хосте, хотя и используют сетевой UDP-канал. При этом UDP-модуль (входит в состав ОС), обнаружив в качестве адреса назначения свой собственный IP-адрес, никакой отправки датаграммы по сети не выполняет, а направляет ее в указанный локальный сокет.(Текст файла file: Osinniki vaz5893)

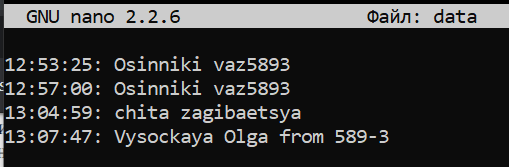
Результат работы программ:



1. Запускаем наш сервер в оперативном режиме и ждем обращения клиентов других пользователей предварительно сообщив им свой порт(1305).

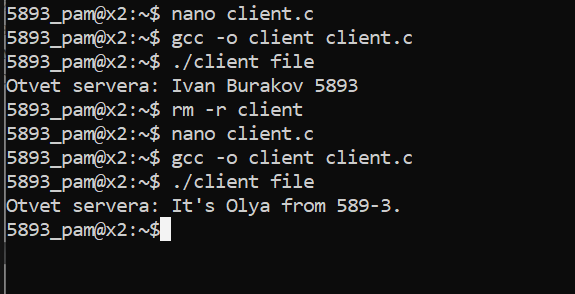
Результат работы сервера:



результирующий файл своей серверной части:

1. Заново компилируем файл client.c меняя поле sin\_port, структуры sockaddr\_in, задавая известные порты других пользователей.

Результат работы клиента:



Вывод: получили навыки создания на основе сокетов датаграммных UDP-каналов, используемых для связи между удаленными процессами.

UDP не требует заранее установленного соединения.