CHAUDHARY HAMDAN

1905387

Networks Lab 5

05/08/2021

1. Write a program to create an UDP socket through which client will send a string to the server and server will echo back the string to the client.

Code (server file):

/\*

\*\* A datagram sockets "server" demo

\*/

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define MYPORT 4952 // the port users will be connecting to

#define MAXBUFLEN 200

int main()

{

int sockfd;

struct sockaddr\_in my\_addr; // my address information

struct sockaddr\_in their\_addr; // connector's address information

socklen\_t addr\_len;

int numbytes;

char buf[MAXBUFLEN];

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) == -1) {

perror("socket");

exit(1);

}

my\_addr.sin\_family = AF\_INET; // host byte order

my\_addr.sin\_port = htons(MYPORT); // short, network byte order

my\_addr.sin\_addr.s\_addr = INADDR\_ANY; // automatically fill with my IP

//memset(my\_addr.sin\_zero, '\0', sizeof my\_addr.sin\_zero);

if (bind(sockfd, (struct sockaddr \*)&my\_addr, sizeof my\_addr) == -1) {

perror("bind");

exit(1);

}

addr\_len = sizeof their\_addr;

if ((numbytes = recvfrom(sockfd, buf, MAXBUFLEN - 1 , 0,

(struct sockaddr \*)&their\_addr, &addr\_len)) == -1) {

perror("recvfrom");

exit(1);

}

printf("got packet from %s\n", inet\_ntoa(their\_addr.sin\_addr));

printf("packet is %d bytes long\n", numbytes);

buf[numbytes] = '\0';

printf("packet contains \"%s\"\n", buf);

sendto(sockfd, buf, strlen(buf), 0, (struct sockaddr \*)&their\_addr, sizeof their\_addr);

close(sockfd);

return 0;

}

Code (client file):

/\*

\*\* A datagram "client" demo

\*/

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <netdb.h>

#define SERVERPORT 4952 // the port users will be connecting to

int main()

{

int sockfd;

struct sockaddr\_in their\_addr; // connector's address information

//struct hostent \*he;

socklen\_t addr\_len;

int numbytes;

char arg[30];

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) == -1) {

perror("socket");

exit(1);

}

their\_addr.sin\_family = AF\_INET; // host byte order

their\_addr.sin\_port = htons(SERVERPORT); // short, network byte order

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

//memset(their\_addr.sin\_zero, '\0', sizeof their\_addr.sin\_zero);

printf("Enter a message\n");

gets(arg);

if ((numbytes = sendto(sockfd, arg, strlen(arg), 0,

(struct sockaddr \*)&their\_addr, sizeof their\_addr)) == -1) {

perror("sendto");

exit(1);

}

printf("sent %d bytes to %s\n", numbytes, inet\_ntoa(their\_addr.sin\_addr));

recvfrom(sockfd, arg, 200 - 1 , 0,

(struct sockaddr \*)&their\_addr, &addr\_len);

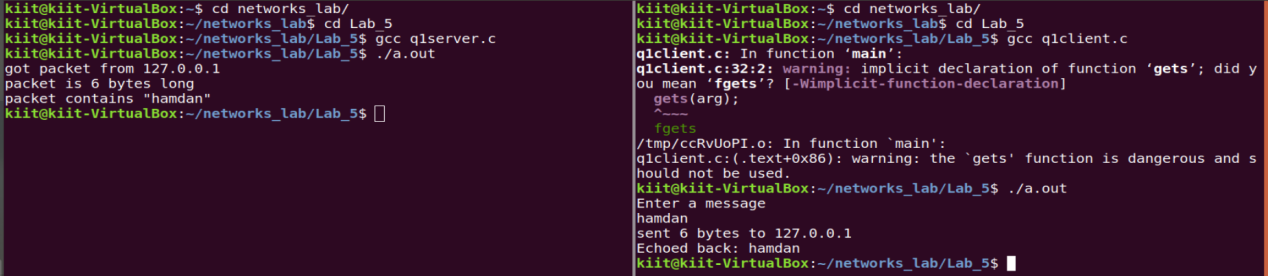
printf("Echoed back: %s\n", arg);

close(sockfd);

return 0;

}

Output:



1. Write a program to create an UDP socket through which client will send an integer number to the server, server will find the sum of digits and return back to the client. And client will display the sum of digits.

Code (server file):

/\*

\*\* A datagram sockets "server" demo

\*/

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define MYPORT 4952 // the port users will be connecting to

#define MAXBUFLEN 200

int main()

{

int sockfd;

struct sockaddr\_in my\_addr; // my address information

struct sockaddr\_in their\_addr; // connector's address information

socklen\_t addr\_len;

int numbytes;

int buf;

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) == -1) {

perror("socket");

exit(1);

}

my\_addr.sin\_family = AF\_INET; // host byte order

my\_addr.sin\_port = htons(MYPORT); // short, network byte order

my\_addr.sin\_addr.s\_addr = INADDR\_ANY; // automatically fill with my IP

//memset(my\_addr.sin\_zero, '\0', sizeof my\_addr.sin\_zero);

if (bind(sockfd, (struct sockaddr \*)&my\_addr, sizeof my\_addr) == -1) {

perror("bind");

exit(1);

}

addr\_len = sizeof their\_addr;

if ((numbytes = recvfrom(sockfd, &buf, sizeof(int) , 0,

(struct sockaddr \*)&their\_addr, &addr\_len)) == -1) {

perror("recvfrom");

exit(1);

}

printf("got packet from %s\n", inet\_ntoa(their\_addr.sin\_addr));

printf("packet is %d bytes long\n", numbytes);

printf("packet contains %d\n", buf);

int sum = 0;

while (buf) {

sum += (buf % 10);

buf /= 10;

}

sendto(sockfd, &sum, sizeof(sum), 0, (struct sockaddr \*)&their\_addr, sizeof their\_addr);

close(sockfd);

return 0;

}

Code (client file):

/\*

\*\* A datagram "client" demo

\*/

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <netdb.h>

#define SERVERPORT 4952 // the port users will be connecting to

int main()

{

int sockfd;

struct sockaddr\_in their\_addr; // connector's address information

//struct hostent \*he;

socklen\_t addr\_len;

int numbytes;

int arg;

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) == -1) {

perror("socket");

exit(1);

}

their\_addr.sin\_family = AF\_INET; // host byte order

their\_addr.sin\_port = htons(SERVERPORT); // short, network byte order

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

//memset(their\_addr.sin\_zero, '\0', sizeof their\_addr.sin\_zero);

printf("Enter a number to find sum of digits: ");

scanf("%d", &arg);

if ((numbytes = sendto(sockfd, &arg, sizeof(int), 0,

(struct sockaddr \*)&their\_addr, sizeof their\_addr)) == -1) {

perror("sendto");

exit(1);

}

printf("sent %d bytes to %s\n", numbytes, inet\_ntoa(their\_addr.sin\_addr));

recvfrom(sockfd, &arg, 200 - 1 , 0,

(struct sockaddr \*)&their\_addr, &addr\_len);

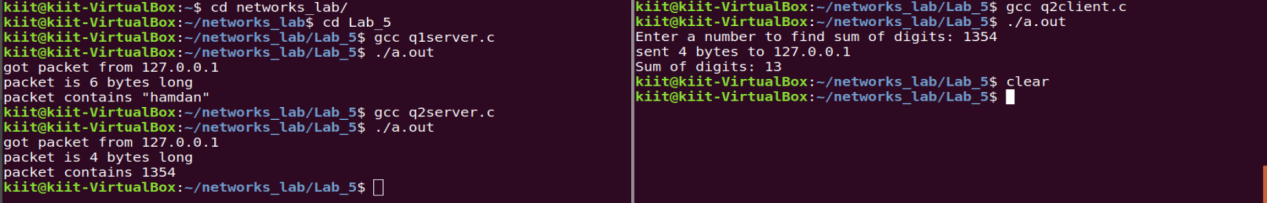
printf("Sum of digits: %d\n", arg);

close(sockfd);

return 0;

}

Output:



1. Write a program to create an UDP socket through which the client will send a key to the server, server already has an integer array stored in ascending order, server will search that key and send the result to the client.

Code (server file):

/\*

\*\* A datagram sockets "server" demo

\*/

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#define MYPORT 4952 // the port users will be connecting to

#define MAXBUFLEN 200

int main()

{

int sockfd;

struct sockaddr\_in my\_addr; // my address information

struct sockaddr\_in their\_addr; // connector's address information

socklen\_t addr\_len;

int numbytes;

int buf;

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) == -1) {

perror("socket");

exit(1);

}

my\_addr.sin\_family = AF\_INET; // host byte order

my\_addr.sin\_port = htons(MYPORT); // short, network byte order

my\_addr.sin\_addr.s\_addr = INADDR\_ANY; // automatically fill with my IP

//memset(my\_addr.sin\_zero, '\0', sizeof my\_addr.sin\_zero);

if (bind(sockfd, (struct sockaddr \*)&my\_addr, sizeof my\_addr) == -1) {

perror("bind");

exit(1);

}

int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

addr\_len = sizeof their\_addr;

if ((numbytes = recvfrom(sockfd, &buf, sizeof(int) , 0,

(struct sockaddr \*)&their\_addr, &addr\_len)) == -1) {

perror("recvfrom");

exit(1);

}

printf("got packet from %s\n", inet\_ntoa(their\_addr.sin\_addr));

printf("packet is %d bytes long\n", numbytes);

printf("packet contains %d\n", buf);

int s = 0, e = 9, ans = -1, flag = 1;

while (s <= e) {

int mid = (s + e) / 2;

if (arr[mid] == buf) {

flag = 0;

sendto(sockfd, &mid, sizeof(mid), 0, (struct sockaddr \*)&their\_addr, sizeof their\_addr);

break;

}

else if (buf < arr[mid]) {

e = mid - 1;

}

else {

s = mid + 1;

}

}

if (flag) {

ans = -1;

sendto(sockfd, &ans, sizeof(ans), 0, (struct sockaddr \*)&their\_addr, sizeof their\_addr);

}

close(sockfd);

return 0;

}

Code (client file):

/\*

\*\* A datagram "client" demo

\*/

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <netdb.h>

#define SERVERPORT 4952 // the port users will be connecting to

int main()

{

int sockfd;

struct sockaddr\_in their\_addr; // connector's address information

//struct hostent \*he;

socklen\_t addr\_len;

int numbytes;

int arg;

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) == -1) {

perror("socket");

exit(1);

}

their\_addr.sin\_family = AF\_INET; // host byte order

their\_addr.sin\_port = htons(SERVERPORT); // short, network byte order

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

//memset(their\_addr.sin\_zero, '\0', sizeof their\_addr.sin\_zero);

printf("Enter a number to search in array: ");

scanf("%d", &arg);

if ((numbytes = sendto(sockfd, &arg, sizeof(int), 0,

(struct sockaddr \*)&their\_addr, sizeof their\_addr)) == -1) {

perror("sendto");

exit(1);

}

printf("sent %d bytes to %s\n", numbytes, inet\_ntoa(their\_addr.sin\_addr));

recvfrom(sockfd, &arg, 200 - 1 , 0,

(struct sockaddr \*)&their\_addr, &addr\_len);

printf("Element found at: %d\n", arg);

close(sockfd);

return 0;

}

Output:

