CHAUDHARY HAMDAN

1905387

Networks Lab 6

12/08/2021

1. Write a c program to create an UDP socket through which client will send the roll no of a student to the server . Then the server already stored 10 students information like name roll no and marks of 5 subjects . Then server will search the roll no in the student list .If it finds the roll no then it will send back the details of the student to the client . And if the roll no is not there , then it sends back " Student not found."

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

#define MAXBUFLEN 200

struct student {

int roll;

char name[10];

int marks[5];

};

int main()

{

int sockfd;

struct sockaddr\_in my\_addr;

struct sockaddr\_in their\_addr;

socklen\_t addr\_len;

int rn;

struct student buf;

struct student arr[10];

int a;

for(a=0;a<10;a++) {

arr[a].roll = a+1;

strcpy(arr[a].name,"Name");

int aa;

for(aa=0;aa<5;aa++)

arr[a].marks[aa] = (a\*10)+aa;

}

int numbytes;

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

perror("socket");

exit(1);

}

my\_addr.sin\_family = AF\_INET;

my\_addr.sin\_port = htons(MYPORT);

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

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

perror("bind");

exit(1);

}

addr\_len = sizeof their\_addr;

if ((numbytes = recvfrom(sockfd, &rn, sizeof(rn), 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", rn);

int flag = 1;

for(a=0;a<10;a++) {

if(arr[a].roll == rn) {

buf = arr[a];

flag = 0;

break;

}

}

if(flag) {

buf.roll = -1;

strcpy(buf.name, "Not Found");

int aa;

for(aa=0;aa<5;aa++)

buf.marks[aa] = -1;

}

sendto(sockfd, &buf, sizeof(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

struct student {

int roll;

char name[10];

int marks[5];

};

int main()

{

int sockfd;

int buf;

struct sockaddr\_in their\_addr;

struct student rec;

socklen\_t addr\_len;

int numbytes;

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

perror("socket");

exit(1);

}

their\_addr.sin\_family = AF\_INET;

their\_addr.sin\_port = htons(SERVERPORT);

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

printf("Enter roll to search: ");

scanf("%d", &buf);

if ((numbytes = sendto(sockfd, &buf, sizeof(buf), 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, &rec, sizeof(rec) , 0,

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

printf("Received Data, roll = %d, name = %s\n", rec.roll, rec.name);

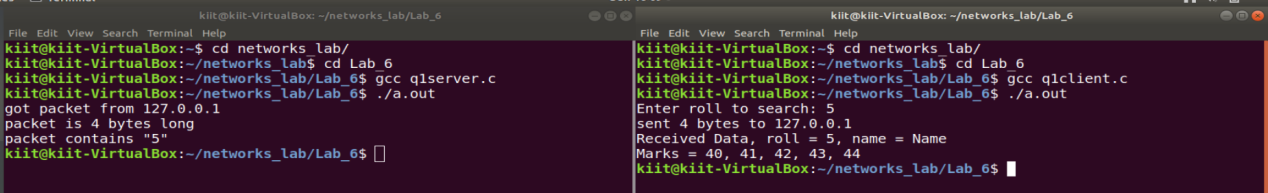
printf("Marks = %d, %d, %d, %d, %d\n", rec.marks[0], rec.marks[1], rec.marks[2], rec.marks[3], rec.marks[4]);

close(sockfd);

return 0;

}

Output:



1. WAP to create an UDP program through which client will send , few commands to the server . Server will respond accordingly. For e.g :

1- If client is sending "list" command, server will send back the list of file names present in the current working directory.

2- If client is sending "retr-filename" command, server will send back the content of that file.

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>

#include <dirent.h>

#define MYPORT 4952

#define MAXBUFLEN 200

int main()

{

int sockfd;

struct sockaddr\_in my\_addr;

struct sockaddr\_in their\_addr;

socklen\_t addr\_len;

char com[30];

char \*rec[10];

int numbytes;

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

perror("socket");

exit(1);

}

my\_addr.sin\_family = AF\_INET;

my\_addr.sin\_port = htons(MYPORT);

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

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

perror("bind");

exit(1);

}

addr\_len = sizeof their\_addr;

if ((numbytes = recvfrom(sockfd, com, sizeof(com), 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 \"%s\"\n", com);

if(strcmp(com, "list")) {

DIR \*d;

struct dirent \*dir;

d = opendir(".");

if (d)

{

int i=-1;

while ((dir = readdir(d)) != NULL)

{

i++;

printf("%s, ", dir->d\_name);

rec[i] = (char \*)dir->d\_name;

}

closedir(d);

}

}

sendto(sockfd, rec, sizeof(rec), 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

int main()

{

int sockfd;

struct sockaddr\_in their\_addr;

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;

their\_addr.sin\_port = htons(SERVERPORT);

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

printf("Enter a command: ");

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));

char \*rec[10][30];

recvfrom(sockfd, rec, sizeof(rec) , 0,

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

printf("Files present: \n");

for(int i=0;i<10;i++) {

printf("%s, ", rec[i]);

}

printf("\n");

close(sockfd);

return 0;

}

Output:

