CHAUDHARY HAMDAN

1905387

Networks Lab 3

22/07/2021

1. Write a C program to create a TCP socket through which client will send a string to the server. The server will echo back the same string to the client. When client sends "exit", then the server will close the connection.

Code (server file):

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

main()

{

int sockfd, fd1, length, i;

char buf[100];

struct sockaddr\_in sa\_addr, cl\_addr;

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

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

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

sa\_addr.sin\_port = htons(6000);

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

i = bind(sockfd, (struct sockaddr \*)&sa\_addr, sizeof(sa\_addr));

printf("test %d%d\n", sockfd, i);

listen(sockfd, 5);

length = sizeof(cl\_addr);

fd1 = accept(sockfd, (struct sockaddr \*) &cl\_addr, &length);

while (1)

{

for (i = 0; i < 100; i++)

buf[i] = '\0';

recv(fd1, buf, 100, 0);

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

if (strcmp(buf, "exit") == 0)

{

break;

}

send(fd1, buf, 100, 0);

}

close(fd1);

}

Code (client file):

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

main()

{

int i, sockfd;

char buf[100];

struct sockaddr\_in sa\_addr;

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

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

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

sa\_addr.sin\_port = htons(6000);

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

i = connect(sockfd, (struct sockaddr \*)&sa\_addr, sizeof(sa\_addr));

while (1)

{

for (i = 0; i < 100; i++)

buf[i] = '\0';

printf("Enter string to send: ");

scanf("%s", &buf);

char temp[100];

strcpy(temp, buf);

send(sockfd, buf, 100, 0);

for (i = 0; i < 100; i++)

buf[i] = '\0';

recv(sockfd, buf, 100, 0);

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

if (strcmp(temp, "exit") == 0)

{

break;

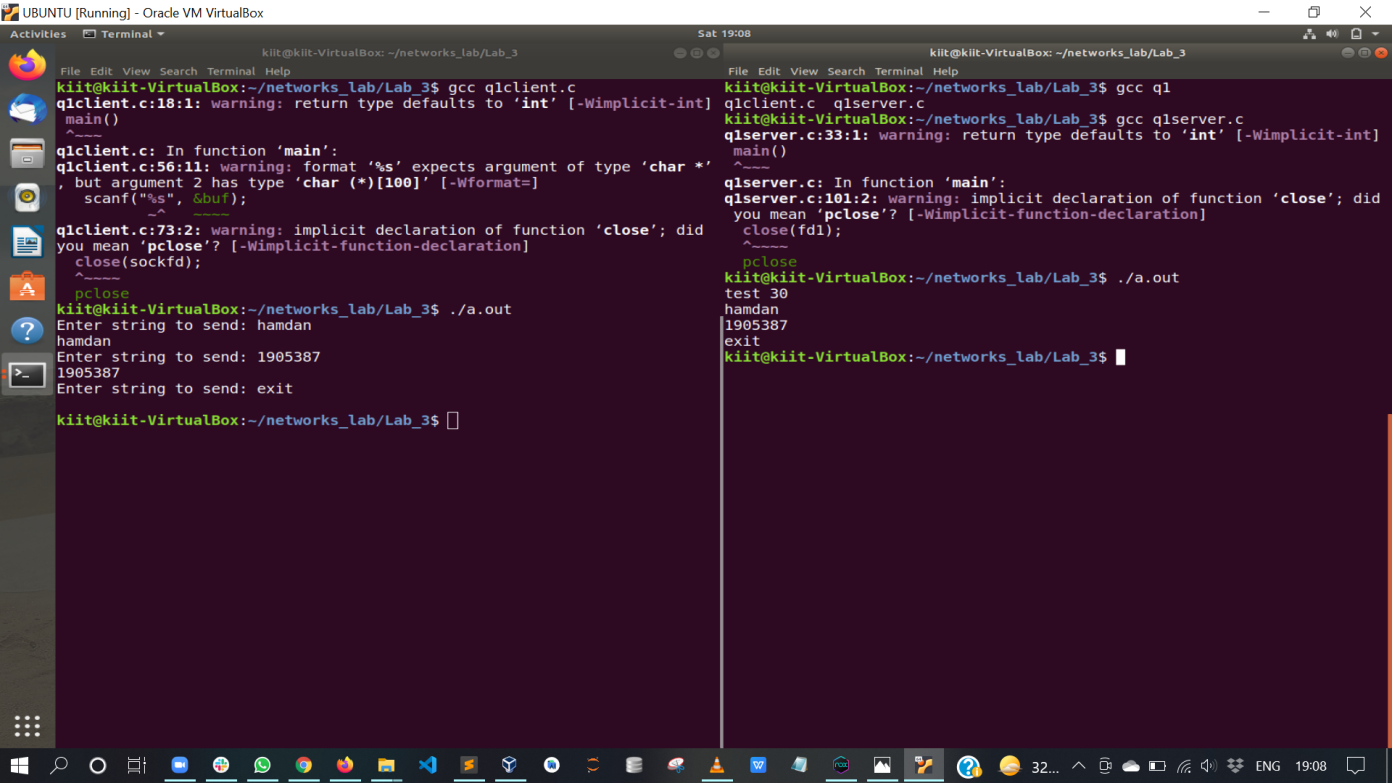
}

}

close(sockfd);

}

Output:



1. Write a C program to create a TCP socket through which client will send an integer number to the server. The server will find the sum of digit(s) of that number and send back to the client and the client will display the sum and the connection will close.

Code (server file):

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

main()

{

int sockfd, fd1, length, i;

int buf;

struct sockaddr\_in sa\_addr, cl\_addr;

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

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

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

sa\_addr.sin\_port = htons(6000);

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

i = bind(sockfd, (struct sockaddr \*)&sa\_addr, sizeof(sa\_addr));

printf("test %d%d\n", sockfd, i);

listen(sockfd, 5);

length = sizeof(cl\_addr);

fd1 = accept(sockfd, (struct sockaddr \*) &cl\_addr, &length);

recv(fd1, &buf, 4, 0);

printf("recv: %d", buf);

int ans = 0;

while (buf)

{

ans += (buf % 10);

buf /= 10;

}

buf = ans;

send(fd1, &buf, 4, 0);

close(fd1);

}

Code (client file):

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

main() {

int i, sockfd;

int buf;

struct sockaddr\_in sa\_addr;

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

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

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

sa\_addr.sin\_port = htons(6000);

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

i = connect(sockfd, (struct sockaddr \*)&sa\_addr, sizeof(sa\_addr));

printf("Enter number: ");

scanf("%d", &buf);

send(sockfd, &buf, 4, 0);

buf = 0;

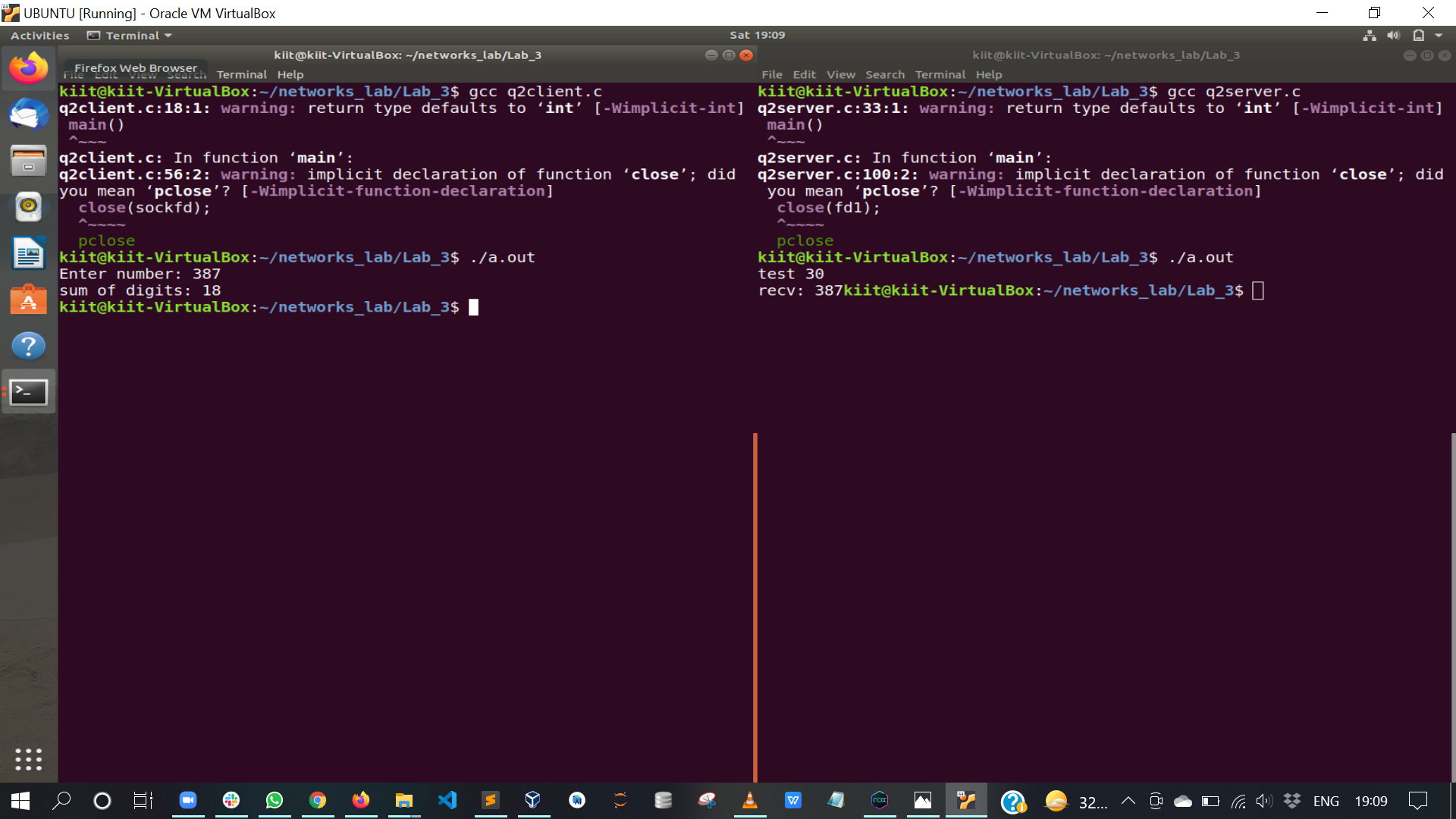
recv(sockfd, &buf, 4, 0);

printf("sum of digits: %d\n", buf);

close(sockfd);

}

Output:



1. Write a C program to create a TCP socket through which the client will send an integer array to the server, the server will sort the array in ascending order and will send back to the client. The clint will display the sorted array and the connection will close.

Code (server file):

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

main()

{

int sockfd, fd1, length, i;

int buf[5];

struct sockaddr\_in sa\_addr, cl\_addr;

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

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

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

sa\_addr.sin\_port = htons(6000);

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

i = bind(sockfd, (struct sockaddr \*)&sa\_addr, sizeof(sa\_addr));

printf("test %d%d\n", sockfd, i);

listen(sockfd, 5);

length = sizeof(cl\_addr);

fd1 = accept(sockfd, (struct sockaddr \*) &cl\_addr, &length);

recv(fd1, buf, 20, 0);

int a, b;

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

{

for (a = b + 1; a < 5; a++)

{

if (buf[b] > buf[a])

{

buf[a] = buf[a] + buf[b];

buf[b] = buf[a] - buf[b];

buf[a] = buf[a] - buf[b];

}

}

}

printf("Sorted array: ");

int ii;

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

{

printf("%d, ", buf[ii]);

}

printf("\n");

send(fd1, buf, 20, 0);

close(fd1);

}

Code (client file):

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

main()

{

int i, sockfd;

int buf[5];

int buff[5];

struct sockaddr\_in sa\_addr;

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

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

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

sa\_addr.sin\_port = htons(6000);

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

i = connect(sockfd, (struct sockaddr \*)&sa\_addr, sizeof(sa\_addr));

printf("Enter 5 numbers: ");

int ii;

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

{

scanf("%d", (buf + ii));

}

send(sockfd, buf, 20, 0);

recv(sockfd, buff, 20, 0);

printf("Sorted array: ");

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

{

printf("%d, ", buff[ii]);

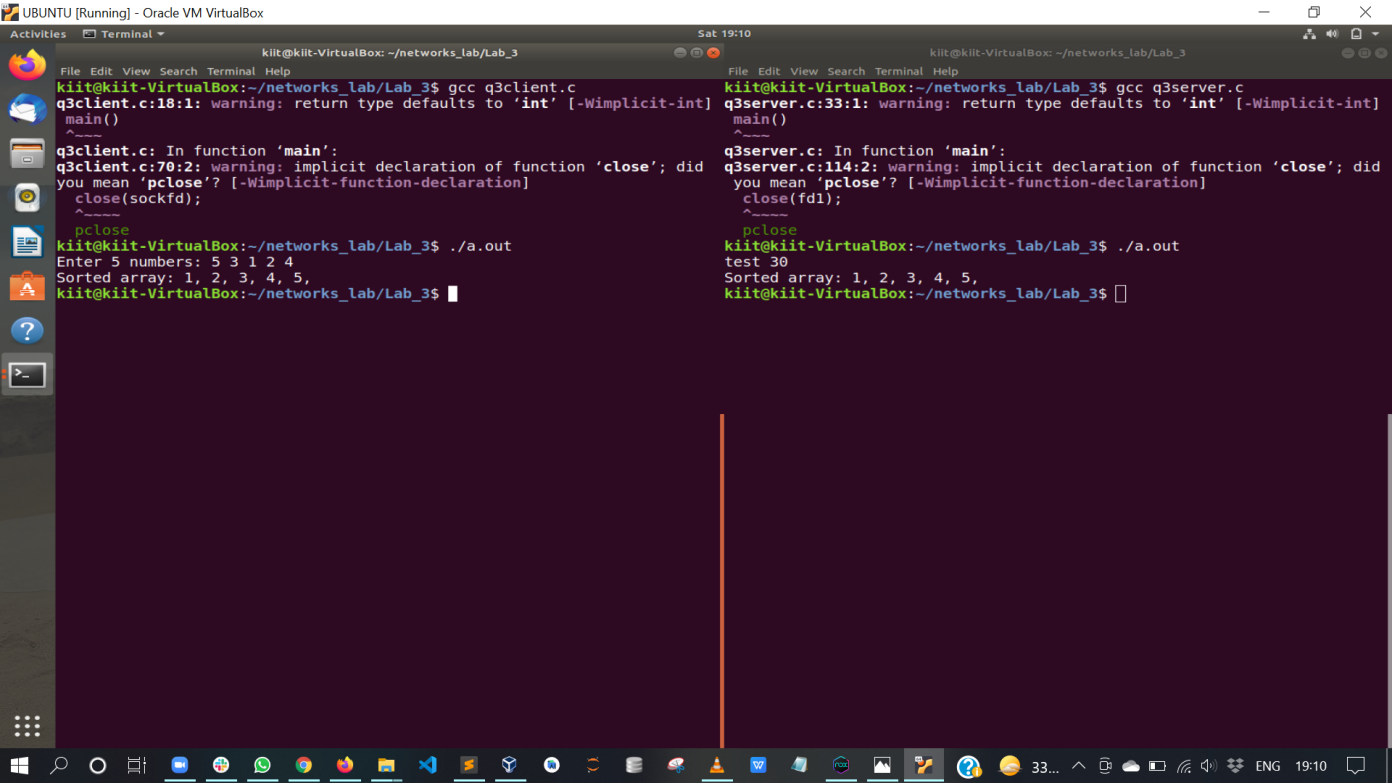
}

printf("\n");

close(sockfd);

}

Output:



1. Write a C program to create a TCP socket through which, two clients will connect to a server. The server will get an integer number from client-1 and will send it to client-2. Then the server will get an integer number from client-2 and will send it to clien-1.

Code (server file):

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

main()

{

int sockfd, fd1, fd2, length, i;

int buf;

struct sockaddr\_in sa\_addr, cl\_addr;

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

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

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

sa\_addr.sin\_port = htons(6000);

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

i = bind(sockfd, (struct sockaddr \*)&sa\_addr, sizeof(sa\_addr));

printf("test %d%d\n", sockfd, i);

listen(sockfd, 5);

length = sizeof(cl\_addr);

fd1 = accept(sockfd, (struct sockaddr \*) &cl\_addr, &length);

fd2 = accept(sockfd, (struct sockaddr \*) &cl\_addr, &length);

recv(fd1, &buf, sizeof(int), 0);

send(fd2, &buf, sizeof(int), 0);

recv(fd2, &buf, sizeof(int), 0);

printf("Received %d in server and sending the same to client 1.\n", buf);

send(fd1, &buf, sizeof(int), 0);

close(fd1);

}

Code (client1 file):

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

main()

{

int i, sockfd;

int buf;

struct sockaddr\_in sa\_addr;

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

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

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

sa\_addr.sin\_port = htons(6000);

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

i = connect(sockfd, (struct sockaddr \*)&sa\_addr, sizeof(sa\_addr));

buf = 1905387;

printf("Sending %d from client 1 to client 2\n", buf);

send(sockfd, &buf, sizeof(int), 0);

buf = 0;

recv(sockfd, &buf, sizeof(int), 0);

printf("Received %d in client 1\n", buf);

close(sockfd);

}

Code (client2 file):

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

main()

{

int i, sockfd;

int buf;

struct sockaddr\_in sa\_addr;

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

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

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

sa\_addr.sin\_port = htons(6000);

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

i = connect(sockfd, (struct sockaddr \*)&sa\_addr, sizeof(sa\_addr));

recv(sockfd, &buf, sizeof(int), 0);

printf("Received %d in client 2 and sending the same to server.\n", buf);

send(sockfd, &buf, sizeof(int), 0);

close(sockfd);

}

Output:

