

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:

```

kiit@kiit-VirtualBox: ~/networks_lab/Lab_3
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc qclient.c
qclient.c:18:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^
qclient.c: In function 'main':
qclient.c:56:11: warning: format '%s' expects argument of type 'char *'
, but argument 2 has type 'char (*)[100]' [-Wformat=]
scanf("%s", &buf);
^
qclient.c:73:2: warning: implicit declaration of function 'close'; did
you mean 'pclose'? [-Wimplicit-function-declaration]
close(sockfd);
^
pclose
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ ./a.out
Enter string to send: hamdan
hamdan
Enter string to send: 1905387
1905387
Enter string to send: exit
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$

kiit@kiit-VirtualBox: ~/networks_lab/Lab_3
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc ql
qlclient.c qlserver.c
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc qlserver.c
qlserver.c:33:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^
qlserver.c: In function 'main':
qlserver.c:101:2: warning: implicit declaration of function 'close'; did
you mean 'pclose'? [-Wimplicit-function-declaration]
close(fd1);
^
pclose
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ ./a.out
test 30
hamdan
1905387
exit
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$

```

2. 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);

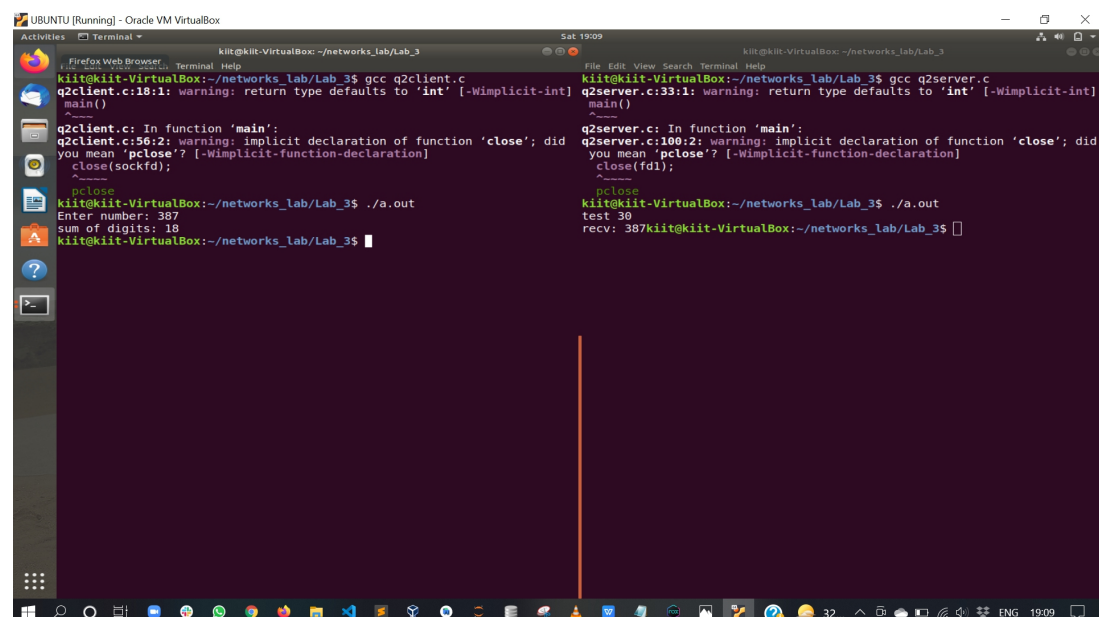
    sa_addr.sin_family = AF_INET;
    sa_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    sa_addr.sin_port = htons(6000);
    memset(sa_addr.sin_zero, '\0', sizeof sa_addr.sin_zero);
    i = connect(sockfd, (struct sockaddr *)&sa_addr, sizeof(sa_addr));

    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:



```
UBUNTU [Running] - Oracle VM VirtualBox
Sat 19:09
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc q2client.c
q2client.c:18:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^~~~~
q2client.c: In function 'main':
q2client.c:36:2: warning: implicit declaration of function 'close'; did
you mean 'pclose'? [-Wimplicit-function-declaration]
    close(sockfd);
    ^~~~~
pclose
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ ./a.out
Enter number: 387
sum of digits: 18
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$

kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc q2server.c
q2server.c:33:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^~~~~
q2server.c: In function 'main':
q2server.c:109:2: warning: implicit declaration of function 'close'; did
you mean 'pclose'? [-Wimplicit-function-declaration]
    close(fd1);
    ^~~~~
pclose
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ ./a.out
test 30
recv: 387kiit@kiit-VirtualBox:~/networks_lab/Lab_3$
```

3. 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 client 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:

```

UBUNTU [Running] - Oracle VM VirtualBox
Sat 19:10

kiit@kiit-VirtualBox: ~/networks_lab/Lab_3
File Edit View Search Terminal Help
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc q3client.c
q3client.c:18:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^~~~~
q3client.c: In function 'main':
q3client.c:70:2: warning: implicit declaration of function 'close'; did
you mean 'pclose'? [-Wimplicit-function-declaration]
    close(sockfd);
    ^~~~~
pclose
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ ./a.out
Enter 5 numbers: 5 3 1 2 4
Sorted array: 1, 2, 3, 4, 5,
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$

kiit@kiit-VirtualBox: ~/networks_lab/Lab_3
File Edit View Search Terminal Help
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc q3server.c
q3server.c:33:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^~~~~
q3server.c: In function 'main':
q3server.c:114:2: warning: implicit declaration of function 'close'; did
you mean 'pclose'? [-Wimplicit-function-declaration]
    close(fd1);
    ^~~~~
pclose
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ ./a.out
test 30
Sorted array: 1, 2, 3, 4, 5,
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$

```


4. 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 client-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:

```

UBUNTU [Running] - Oracle VM VirtualBox
Activities Terminal
kiit@kiit-VirtualBox: ~/networks_lab/Lab_3
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc q4client1.c
q4client1.c:18:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^
q4client1.c: In function 'main':
q4client1.c:59:2: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
close(sockfd);
^
pclose
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ ./a.out
Sending 1905387 from client 1 to client 2
Received 1905387 in client 1
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$

kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc q4server.c
q4server.c:33:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^
q4server.c: In function 'main':
q4server.c:92:2: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
close(fd1);
^
pclose
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ ./a.out
test 30
Received 1905387 in server and sending the same to client 1.
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$

kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ gcc q4client2.c
q4client2.c:18:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^
q4client2.c: In function 'main':
q4client2.c:54:2: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
close(sockfd);
^
pclose
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$ ./a.out
Received 1905387 in client 2 and sending the same to server.
kiit@kiit-VirtualBox:~/networks_lab/Lab_3$

```