Socket Programming

USER DATAGRAM PROTOCOL

```c
// udps.c

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>

/*
    socket()
    fill server and client fields
    bind()
    recvfrom()
*/

void main() {
    struct sockaddr_in server, client;
    char buffer[100];
    int sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    client.sin_family = AF_INET;
    client.sin_addr.s_addr = INADDR_ANY;
    client.sin_port = 8080;
    socklen_t len = sizeof(client);
    bind(sockfd, (struct sockaddr*)&server, sizeof(server));
    recvfrom(sockfd, buffer, 100, 0, (struct sockaddr*)&client, &len);
    printf("Message received is: %s", buffer);
}

// udpc.c

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

```c
/*
    socket()
    fill server fields
    sendto()
*/

void main() {
    struct sockaddr_in server;
    char buffer[100];
    int sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    printf("Enter message to be sent: ");
    fgets(buffer, 100, stdin);
    sendto(sockfd, buffer, 100, 0, (struct sockaddr*)&server, sizeof(server));
}
```

```
OUTPUT

sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ gcc udps.c -o server
sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ ./server
Message received is: hello

sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ gcc udpc.c -o client
sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ ./client
Enter message to be sent: hello
```

TRANSMISSION CONTROL PROTOCOL

```c
// tcps.c

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>

/*
    socket()
    fill server and client fields
    bind()
    listen()
    accept()
    recv()
    close()
*/

void main() {
    struct sockaddr_in server, client;
    char buffer[100];
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    client.sin_family = AF_INET;
    client.sin_addr.s_addr = INADDR_ANY;
    client.sin_port = 8080;
    bind(sockfd, (struct sockaddr*)&server, sizeof(server));
    listen(sockfd, 5);
    socklen_t len = sizeof(client);
    int clientfd = accept(sockfd, (struct sockaddr*)&client, &len);
    recv(clientfd, buffer, 100, 0);
    printf("Message received is: %s", buffer);
    close(clientfd);
}


// tcpc.c

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>

/*
    socket()
    fill server fields
    connect()
    send()
    close()
*/

void main() {
    struct sockaddr_in server;
    char buffer[100];
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    connect(sockfd, (struct sockaddr*)&server, sizeof(server));
    printf("Enter message to be sent: ");
    fgets(buffer, 100, stdin);
    send(sockfd, buffer, 100, 0);
    close(sockfd);
}
```

```
OUTPUT

sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ gcc tcps.c -o server
tcps.c: In function ‘main’:
tcps.c:31:5: warning: implicit declaration of function ‘close’; did you mean ‘pc
   31 |     close(clientfd);
      |     ^~~~~
      |     pclose
sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ ./server
Message received is: hello

sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ gcc tcpc.c -o client
tcpc.c: In function ‘main’:
tcpc.c:24:5: warning: implicit declaration of function ‘close’; did you mean ‘pc
   24 |     close(sockfd);
      |     ^~~~~
      |     pclose
sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ ./client
Enter message to be sent: hello
```

FILE TRANSFER PROTOCOL

```
IMPORTANT - WILL WORK ONLY IF THE FILE HAS A SINGLE LINE OF TEXT.


// ftps.c

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>

/*
    socket()
    fill server and client fields
    bind()
    listen()
```

```c
    accept()
    receive file contents
    close()
*/

void main() {
    struct sockaddr_in server, client;
    char name[100], contents[100];
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    client.sin_family = AF_INET;
    client.sin_addr.s_addr = INADDR_ANY;
    client.sin_port = 8080;
    socklen_t len = sizeof(client);
    bind(sockfd, (struct sockaddr*)&server, sizeof(server));
    listen(sockfd, 5);
    int clientfd = accept(sockfd, (struct sockaddr*)&client, &len);
    recv(clientfd, name, 100, 0);
    printf("Name of the file received: %s\n", name);
    recv(clientfd, contents, 100, 0);
    printf("The file contents: %s\n", contents);
    close(clientfd);
}

// ftpc.c

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>

/*
    socket()
    fill server fields
    connect()
    send file contents
    close()
*/

void main() {
    struct sockaddr_in server;
    char name[100], contents[100];
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    connect(sockfd, (struct sockaddr*)&server, sizeof(server));
    printf("Enter the name of the file: ");
    gets(name); // do not use fgets() here
    send(sockfd, name, 100, 0);
    FILE *fp = fopen(name, "r");
    fgets(contents, 100, fp);
    send(sockfd, contents, 100, 0);
    close(sockfd);
}
```

```
OUTPUT

sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ gcc ftps.c -o server
ftps.c: In function ‘main’:
ftps.c:33:5: warning: implicit declaration of function ‘close’; did you mean ‘pc
   33 |     close(clientfd);
      |     ^~~~~
      |     pclose
sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ ./server
Name of the file received: sample.txt
The file contents: hello, everyone!

sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ gcc ftpc.c -o client
ftpc.c: In function ‘main’:
ftpc.c:22:5: warning: implicit declaration of function ‘gets’; did you mean ‘fge
   22 |     gets(name);
      |     ^~~~
      |     fgets
ftpc.c:27:5: warning: implicit declaration of function ‘close’; did you mean ‘pc
   27 |     close(sockfd);
      |     ^~~~~
      |     pclose
/usr/bin/ld: /tmp/ccGZEEUn.o: in function `main':
ftpc.c:(.text+0x91): warning: the `gets' function is dangerous and should not be
sanbabyfrancis@DESKTOP-F4PQ7B4:/programs/network$ ./client
Enter the name of the file: sample.txt
```