File Transfer-UDP

Input :

udpFTserver.c

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<netdb.h>
#include<unistd.h>

int main()
{
	//Variables
	FILE *fp;
	int sock,pt,cnt=0;
	char filename[1024],buf_recv[1024],buf_send[1024];
	struct sockaddr_in server,client;
	int slen = sizeof(client);

	//Creating a Socket
	sock = socket(AF_INET,SOCK_DGRAM,0);
	if(sock == -1)
	{
		perror("Socket Error");
		exit(1);
	}
	server.sin_family = AF_INET;
	server.sin_addr.s_addr = INADDR_ANY;
	server.sin_port = htons(8888);

	//Binding socket to IP and Port
	if(bind(sock, (struct sockaddr *)&server, sizeof(server)) == -1)
	{
		perror("Bind Error");
		exit(1);
	}

	while(1)
	{
		cnt=0;
```

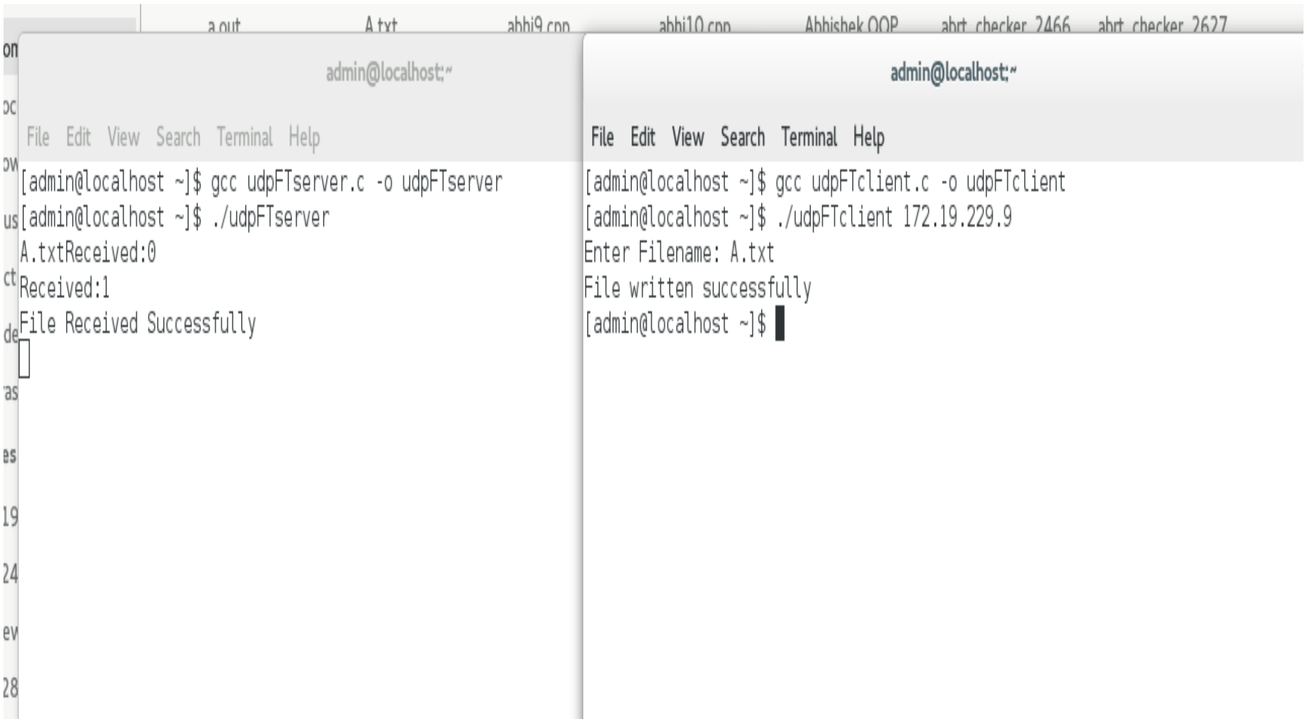
```
        fflush(stdout);
        pt = recvfrom(sock, filename, sizeof(filename), 0, (struct sockaddr *)&client,
&slen);
        printf("%s",filename);

        //strcat(filename,"1");
        fp=fopen(filename, "w");

        //Initialize filedata with some random value
        strcpy(buf_recv,"random");

        //Receiving file data in packets till end of file
        while(strcmp(buf_recv,"end") != 0)
        {
            //Receiving File data
            /***********Definerecvfrom in buf_recv Here**********/
            recvfrom(sock, buf_recv, sizeof(buf_recv), 0, (struct sockaddr *)&client,
&slen);
            //Writing received file data
            /***********Definefwrite here Here**********/
            if(strcmp(buf_recv,"end") != 0)
                    {
                    fprintf(fp,buf_recv);
                    }
            printf("Received:%d\n",cnt);
            cnt++;
        }
        //Closing the file
        fclose(fp);
        printf("File Received Successfully\n");
    }
}
```

udpFTclient.c

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<netdb.h>
#include<unistd.h>

//Check if the File Exists
int exists(const char *fname)
```

```
{
   FILE *file;
if(file = fopen(fname, "r"))
   {
fclose(file);
return 1;
   }
return 0;
}

int main(int argc, char* argv[])
{
        //Variables
        FILE *fp;
        int sock,pt;
        char filename[1024],buf_recv[1024],buf_send[1024];
        struct sockaddr_in server;
        struct hostent *host;
        int slen = sizeof(server);

        //Taking the Commanf Line Argument of IP address
        host = gethostbyname(argv[1]);

        //Creating a Socket
        sock = socket(AF_INET,SOCK_DGRAM,0);
        if(sock == -1)
        {
                perror("Socket Error");
                exit(1);
        }
        server.sin_family = AF_INET;
        server.sin_port = htons(8888);
        memcpy(&server.sin_addr,host->h_addr,host->h_length);

        //Scanning the File Name
        printf("Enter Filename: ");
        scanf("%s",filename);

        //Check if the File Exists
        if(exists(filename))
        {
                //Sending the FileName to the server
                sendto(sock, filename, sizeof(filename), 0, (struct sockaddr *)&server, slen);

                //Opening the file in read mode
                fp=fopen(filename, "rb");
                usleep(100000);
                //Reading the file in chunks
                while (fread(buf_send, strlen(buf_send)+1, 1, fp) == 1)
                {
                    //Sending the read file chunk to the server
                    /***********Definesendto Here**********/
                    sendto(sock, buf_send, sizeof(buf_send), 0, (struct sockaddr *)&server,
slen);
                    usleep(1000);
                }
                usleep(100000);
                //Checking End of File
                if (feof(fp))
                {
                    //When end of file, sending the last chunk of data
                  /***********Definesendto Here**********/
                sendto(sock, buf_send, sizeof(buf_send), 0, (struct sockaddr *)&server, slen);
                  //Sending "end" as data in last packet indicating end of file data
                    strcpy(buf_send,"end");
                    /***********Definesendto Here**********/
                    sendto(sock, buf_send, sizeof(buf_send), 0, (struct sockaddr *)&server,
slen);
                printf("File written successfully\n");
                }
                else
                {
                    printf("File not read/written successfully\n");
                }
                //Closing the file
                fclose(fp);

        }
        else
        {
                printf("File Doesn't Exist\n");
        }

}
```

Output:



```
[admin@localhost ~]$ gcc udpFTserver.c -o udpFTserver
[admin@localhost ~]$ ./udpFTserver
A.txtReceived:0
Received:1
File Received Successfully
```

```
[admin@localhost ~]$ gcc udpFTclient.c -o udpFTclient
[admin@localhost ~]$ ./udpFTclient 172.19.229.9
Enter Filename: A.txt
File written successfully
[admin@localhost ~]$
```