

FTP_CLIENT

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>
#include<arpa/inet.h>
int main ()
{
    FILE *fp;
    int csd,n,ser,s,cli,cport,newst;
    char name[100],rcvg[100],fname[100];
    struct sockaddr_in servaddr;
    printf("enetr the port");
    scanf("%d",&cport);
    csd=(socket(AF_INET,SOCK_STREAM,0));
    if (csd<0)
    {
        perror("socketerror");
        exit(1);
    }
    else
    printf("socket is created");
    servaddr.sin_family=AF_INET;
    //servaddr.sin_addr.s_addr= htonl(INADDR_ANY);
    inet_pton(AF_INET,"127.0.0.1",&servaddr.sin_addr);
    servaddr.sin_port=htons(cport);
    if (connect(csd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
    {
        printf("error in connection");
        exit(1);
    }
    else{
        printf("CONNECTED \n");
        printf("enter existing file name/t");
        scanf("%s",name);
        printf("enetr new file name)\t");
        scanf("%s",fname);
        fp=fopen(fname,"w");
        send(csd,name,sizeof(name),0);
        while(1)
        {
            s=recv(csd,rcvg,100,0);
            rcvg[s]='\0';
            if(strcmp(rcvg,"error")==0)
            {
                printf("File not available");
                break;
            }
            if (strcmp(rcvg,"completed")==0)
            {
                printf("file is transmitted");
                fclose(fp);
                close(csd);
                break;
            }
        }
    }
}
```

```

else
fputs(rcvg,stdout);
printf("%s",rcvg);
return 0;
}
}
FTP_SERVER
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<string.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<fcntl.h>
int main ()
{
FILE *fp;
int sd,newsd,scr,n,a,cli,pid,bd,port,clilen;
char name [100],fileread[100],fname[100],ch,file[100],rcv[100];
struct sockaddr_in servaddr,cliaddr;
printf("enetr port adress");
scanf("%d",&port);
sd=socket(AF_INET,SOCK_STREAM,0);
if(sd<0)
{
printf("cant created,\n");
exit(1);
}
else
printf("socket is created\n");
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
servaddr.sin_port=htons(port);
a=sizeof(servaddr);
bd=bind(sd,(struct sockaddr*)&servaddr,a);
if(bd<0)
{
printf("cannot bind \n");
exit(1);
}
else
printf("binded\n");
listen(sd,5);
clilen=sizeof(cliaddr);
newsd=accept(sd,(struct sockaddr*)&cliaddr,&clilen);
if (newsd<0)
{
printf("Cant accept");
}
else
printf("accepted");
n=recv(newsd,rcv,100,0);
rcv[n]='\0';
fp=fopen(rcv,"r");
if(fp==NULL)
{

```

```
send(newsd,"error",5,0);
close(newsd);
}
else

while(fgets(fileread,sizeof(fileread),0)<0)
{
if(send(newsd,fileread,sizeof(fileread),0)<0)
{
printf("cant dent dile conetent");
}sleep(1);
}
if(!fgets(fileread,sizeof(fileread),fp))
{
send(newsd,"completed",999999999,0);
}
return(0);
}
```