How to run:

1. First compile and run the server file in SERVER directory. gcc server.c ./a.out
2. Compile and run the Client file in CLIENT directory. gcc Client.c ./a.out

Logic:

Firstly since I implemented this to be command line argument code. I will know the number of arguments given and also what they are. So I iterated through all the files and processes them one by one. Since the first argument is ./a.out I skipped it and furhter any command will be treated equivalent to a file. Keeping this in mind I iterated through all the files, and the first thing I would check is if there is already a file with same name exisiting in the current directory. For knowing the existence of the file in the directory I used access funtion which does exactly what I need, F_OK will check if there is a file eith the passed name, W_OK will chakc if the given file exists has the write permissions and many more interesting applications. Using this in-built function I was able to check if the file already exists, As a design choice I asked the user if he wants to overwrite the contents (Note that no more options are provided as a design choice.). This part os coded as below:

```
    for(int i = 1 ;i < argc ; i++)                    // Iterating
through indices of the file
    {
        // printf("%s\n",argv[i]);
        if(access(argv[i],F_OK)==-1)                  // Checking if
there is a file with same name.
        {
            copythecontent(argv[i]);
        }
        else
        {
            printf("%s : already exists\n",argv[i]);
            printf("Press 1 if you want to overwrite %s any other option
will automatically lead it to skip this file\n",argv[i]);
            int num;
            scanf("%d",&num);                         // This is the
response of the user.
            if(num!=1)
            {
                printf("Skipping the file: %s .......\n",argv[i]);
            }
            else
            {
                copythecontent(argv[i]);
            }

        }
    }
```

For checking whether all the files are done processinf or not, I passed a send to server "NOMORETRINADH" , an assumption is that this is not a file name, I used this variable to be the string which says that there are

no more files which are to be processed.

```
memset(&buffer,'\0',bufflen);                        // Once iterating
through all the file is done.
    strcpy(buffer,"NOMORETRINADH");                      // I will send a
signal to server that all the  files are done. SO that server will know
that this clients requirements are done.
    if(send(sock,buffer,bufflen,0)<0)                  // checking if the
send failed.
    {
        printf("Signal that there are no more files left is failed to
send\n");
        perror("send failed");
    }
    else
    {
        printf("Signal that there are no more files left to get copied is
sent to the server\n");
    }
```

Before each send and recv I checked its return value and kept appropriate print and perror statemants so that we can ensure that there is proper error handling which is taken care by this. Also I tried to open the file in the client side with appropriate flags if it fails then I sent the same using a string I communicated with the server, When server sees this it will immediately return from the task of further continuing.

```
if(fd==NULL)                                        // is no file then we
will create one with the same
    {
                                                        // name.
        printf("\n%s : failed to open the required file\n",arg);
        perror("fopen failed");

        memset(&buffer,'\0',bufflen);
        strcpy(buffer,"FILENO");
        if(send(sock,buffer,bufflen,0)<0)
        {
            printf("\n%s : Failed to send server that this file is failed
to open\n",arg);
            perror("send failed");
        }
        return -1;
    }
```

Once we verify if there is the file is open then I just copied the file contents as recivd from the server accordingly. This is implemented in somewhat similar to assignment logic. In the sever side I kept recv sattement which waits for client to send the file name. If a particualr string NOMORETRINADH is sent then there are no more files for the current client in which case I will terminate immediately. In other cases I will process the further info regarding the file.Checkstatus function will take care as to wether the file can be

read or not, if this is not the case then we return -1, so the place where we called this function we will terminate this clients further procesin gfor that particular file.

```
int checkfilestatus()
{
    if(access(buff,F_OK)==-1)
    {
        printf("%s : No such file exists\n",buff);
        return -1;
    }
    if(access(buff,R_OK)==-1)
    {
        printf("%s : Write permission is not there\n",buff);
        return -1;
    }
    return 0;
}
```

Then we will pass the size of the asked file to the client this thing we got using the stat of the file, Similar to the clinet side the server will this time recive the signal sent by the client which will be sent FILENO if it failed to open file in client directory, or else it will send the FILEYES which says that there is a file which does the exact opposite.

```
      if(recv(new_socket,buffer,bufflen,0)<0)
        {
            perror("send failed");
            return -1;
        }
        if(strcmp("FILENO",buffer)==0)
        {
            printf("%s : Was not able to open in client side\n",buff);
            return -1;
        }
        else if(strcmp("FILEYES",buffer)==0)
        {
         // this is left wantedly which says that there is nothing to do if
we were to open the file in the client side directory.
        }
        else
        {
            printf("OOPS!! Didnt expect this message to be passed!\n");
            return -1;
        }
```

Once this part is done it is cake walk we will open the file using fopen and then we will go thropugh the file and will send the corresponding blocks of content to the client side. I also printed the percentage in this side as well. I also checked the final acknowledgment which will verify if entire file is transferred or not.

ASSUMPTIONS:

If the file requested by client is already there in the client side then the permissions of the same will be applying i.e if there is no write permission, then no further copying stage will be initiated, this is in keeping view that the files permissions shouldnt be changed.

Ctrl C will exit the running program both in server and client and corresponding messages are printed.

It will take a while after the downloading is complete please wait for that extra time.

If you want to increse or decrese file transmission rate please change bufflen variable in client and server files.