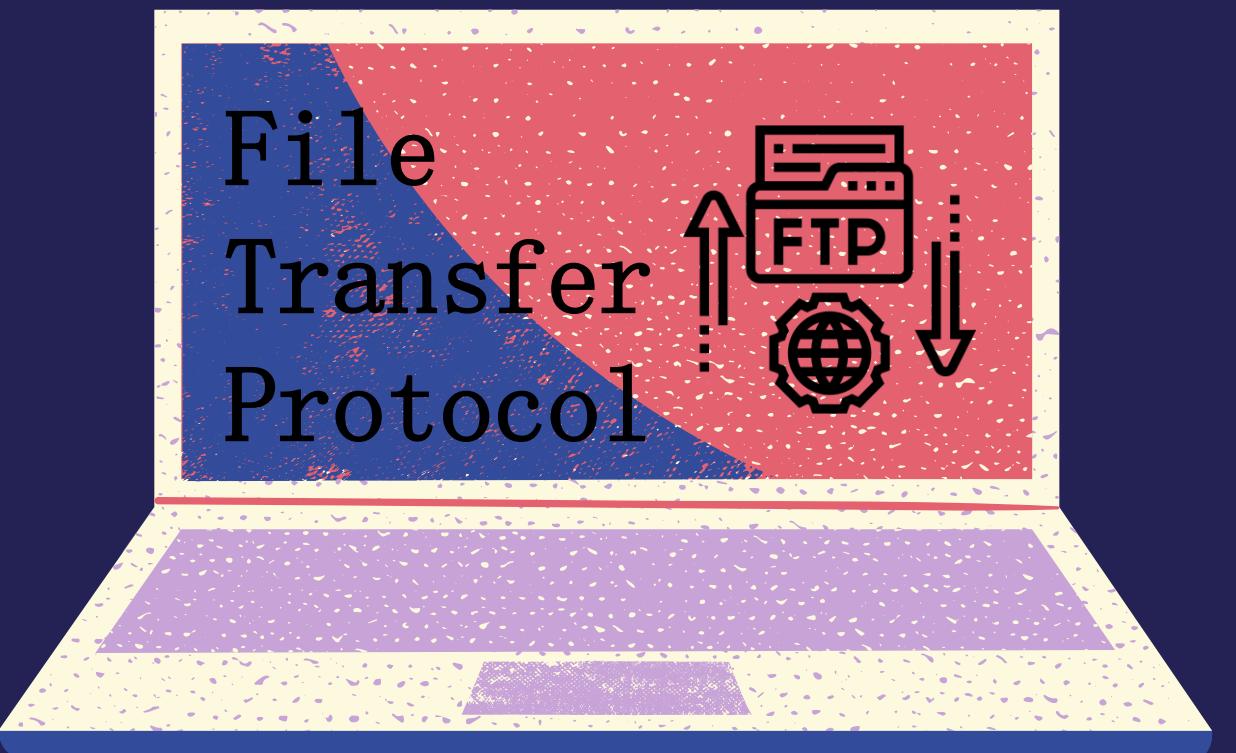


File Transfer Protocol

A standard network protocol used to transfer files between computers over Transmission Control Protocol/Internet Protocol (TCP/IP)connections



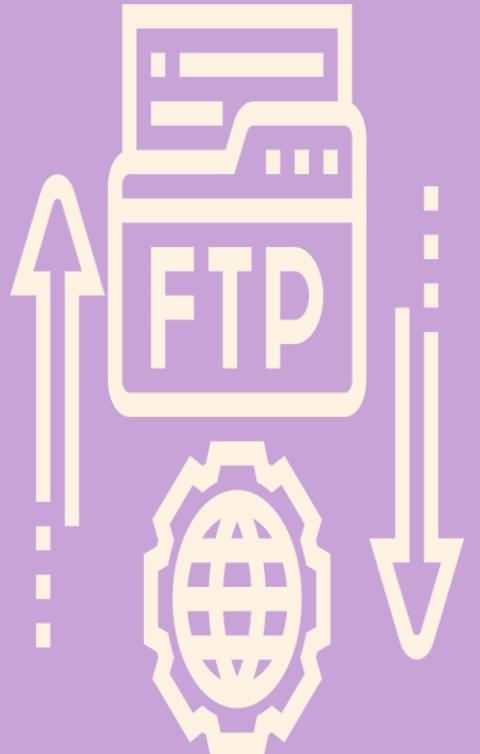


1. Introduction to FTP
2. FTP Architecture
3. FTP IN RFC 959 Architecture
4. FTP Advantages
5. FTP Disadvantages
6. FTP Flowchart
7. Recent use of protocol
8. Implementation Details
9. Execution Screenshots
10. Conclusion

Introduction to FTP

FTP, or File Transfer Protocol, is famous for being a standard network protocol used to transfer files between computers over (TCP/IP) connections. It is a reliable and secure way to transfer files, and it is often used to backup data, upload files to websites, and download files from the internet.

- ❖ It is a simple protocol to use. FTP uses a client-server architecture, which makes it easy to understand and use. There are many FTP clients available for different operating systems, so it is easy to find a client that works for you.
- ❖ It is a reliable protocol. FTP has been around for many years, and it is a well-established protocol. This means that it is reliable and unlikely to experience problems.
- ❖ It is a secure protocol. FTP can be used to transfer files securely, using a variety of security protocols. This makes it a good choice for transferring sensitive data.
- ❖ Backing up data. FTP is a popular way to backup data because it is a reliable and secure way to transfer files.
- ❖ Uploading files to websites. FTP is often used to upload files to websites, such as images, videos, and documents.
- ❖ Downloading files from the internet. FTP can also be used to download files from the internet, such as software, music, and movies.
- ❖ Transferring files between computers. FTP can be used to transfer files between any two computers that are connected to the internet.



The client and server communicate with each other using two separate connections:

1. Control Transfer
2. Data transfer.

The data connection is used to transfer the actual files. The control connection is typically port 21, and the data connection is a dynamic port that is chosen by the server. The client and server communicate using a series of commands and responses.

The most common commands are:

- ❖ PORT: This command tells the server the IP address and port number of the client's data connection.
- ❖ LIST: This command lists the files on the server.
- ❖ RETR: This command retrieves a file from the server.
- ❖ STOR: This command stores a file on the server.

This architecture is a simple and efficient way to transfer files between computers. It is a well-established protocol that has been around for many years. This makes it a reliable and secure way to transfer files.

- It is simple to understand and use. This makes it a good choice for users who are not familiar with networking protocols.
- It is efficient in terms of both bandwidth and CPU usage. This makes it a good choice for transferring large files.
- A reliable way to transfer files. This is because it uses a well-established protocol and a secure method of communication.
- It can be used to transfer files securely. This is because it supports a variety of security protocols, such as SSL and TLS.



FTP in Request For Comments 959 Architecture.

RFC is a document that describes an Internet standard, protocol, or architecture. RFCs are published by the IETF, which is a group of volunteers who develop and maintain the Internet's technical standards.

The original specification for the File Transfer Protocol (FTP) was published in RFC 354 in 1971. This RFC was later superseded by RFC 959, which is the current standard for FTP. RFC 959 describes the syntax and semantics of the FTP protocol, as well as the commands and responses that are used to transfer files between hosts.

RFCs are an important part of the Internet's technical infrastructure. They provide a way for the IETF to document and share information about Internet standards. RFCs are also used to define the behavior of Internet applications and services.

In the context of FTP, RFCs are used to define the syntax and semantics of the FTP protocol, as well as the commands and responses that are used to transfer files between hosts. RFCs are also used to document the security considerations for FTP, as well as the best practices for using FTP.

Here are some of the most important RFCs related to FTP:

RFC 354: File Transfer Protocol

RFC 959: File Transfer Protocol

RFC 1738: Uniform Resource Locators (URL)

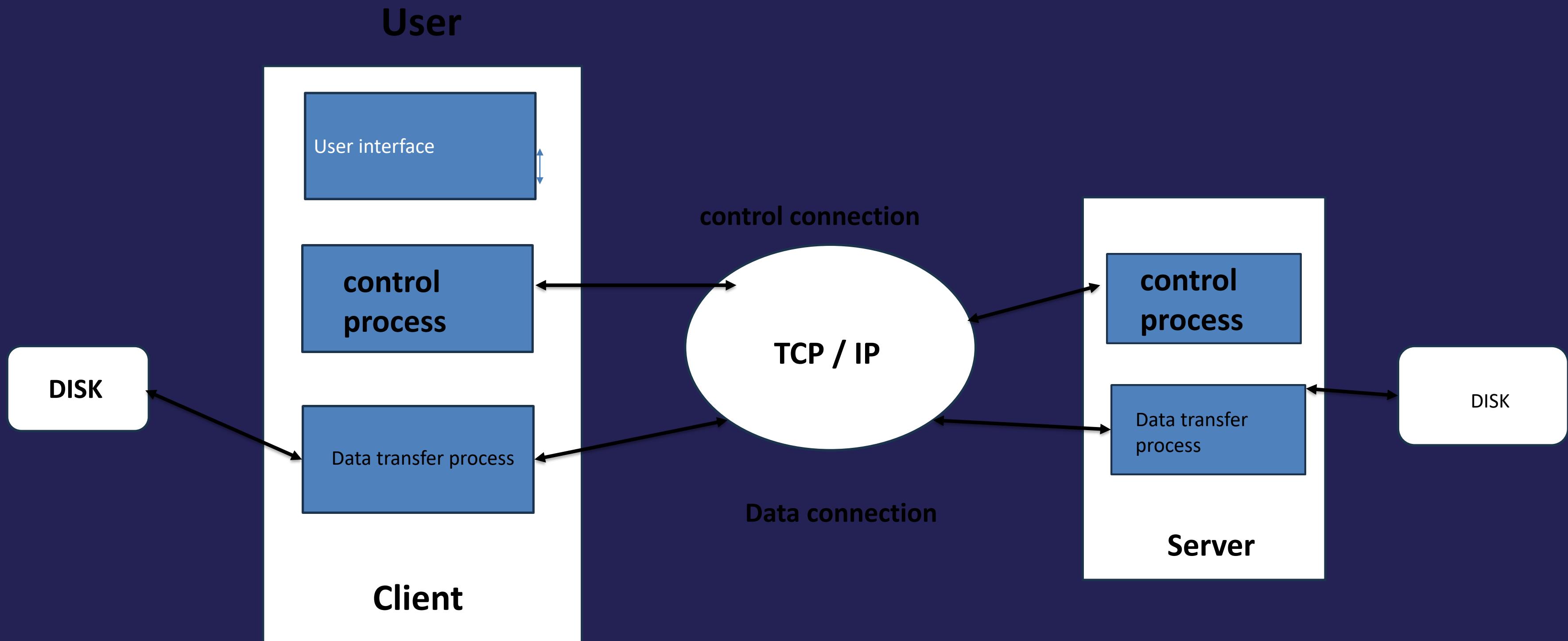
RFC 2228: FTP Security Extensions

RFC 2389: FTP Extensions for IPv6 and NAT



FTP Flow Chart

Mechanism of File Transfer Protocol



Advantages

❖ Reliable and secure:

FTP uses a connection-oriented mode, which means that the connection between the client and server is maintained for the duration of the transfer. This makes it less likely that the transfer will be interrupted. FTP also supports a variety of security features, such as user authentication and encryption, to protect the data being transferred.

❖ Versatile:

FTP can be used to transfer files between any two computers that are connected to a network. This makes it a valuable tool for transferring files between different operating systems and platforms.

❖ Simple to use:

There are many FTP clients available for both Windows and macOS, and most of them are easy to learn and use.

❖ Supports large file transfers:

FTP can typically transfer files faster than other protocols, such as HTTP. This makes it a good choice for transferring large files, such as videos, audio files, and software applications.

❖ Automate file transfers:

FTP can be used to automate file transfers, which can save time and effort. For example, you could set up an FTP client to automatically download files from a remote server at a specific time each day.

❖ Overall, FTP is a reliable, secure, and versatile protocol that is well-suited for transferring files between computers. It is a simple protocol to use and can be used to transfer large files quickly. FTP can also be used to automate file transfers, which can save time and effort.

Disadvantages of the FTP

❖ Security:

FTP uses clear text authentication, which means that passwords are sent over the network in plain text. This can make it vulnerable to eavesdropping.

❖ Complexity:

FTP can be a bit complex to use, especially for beginners.

❖ Lack of features:

FTP does not support some features that are available in other file transfer protocols, such as file versioning and checksums.

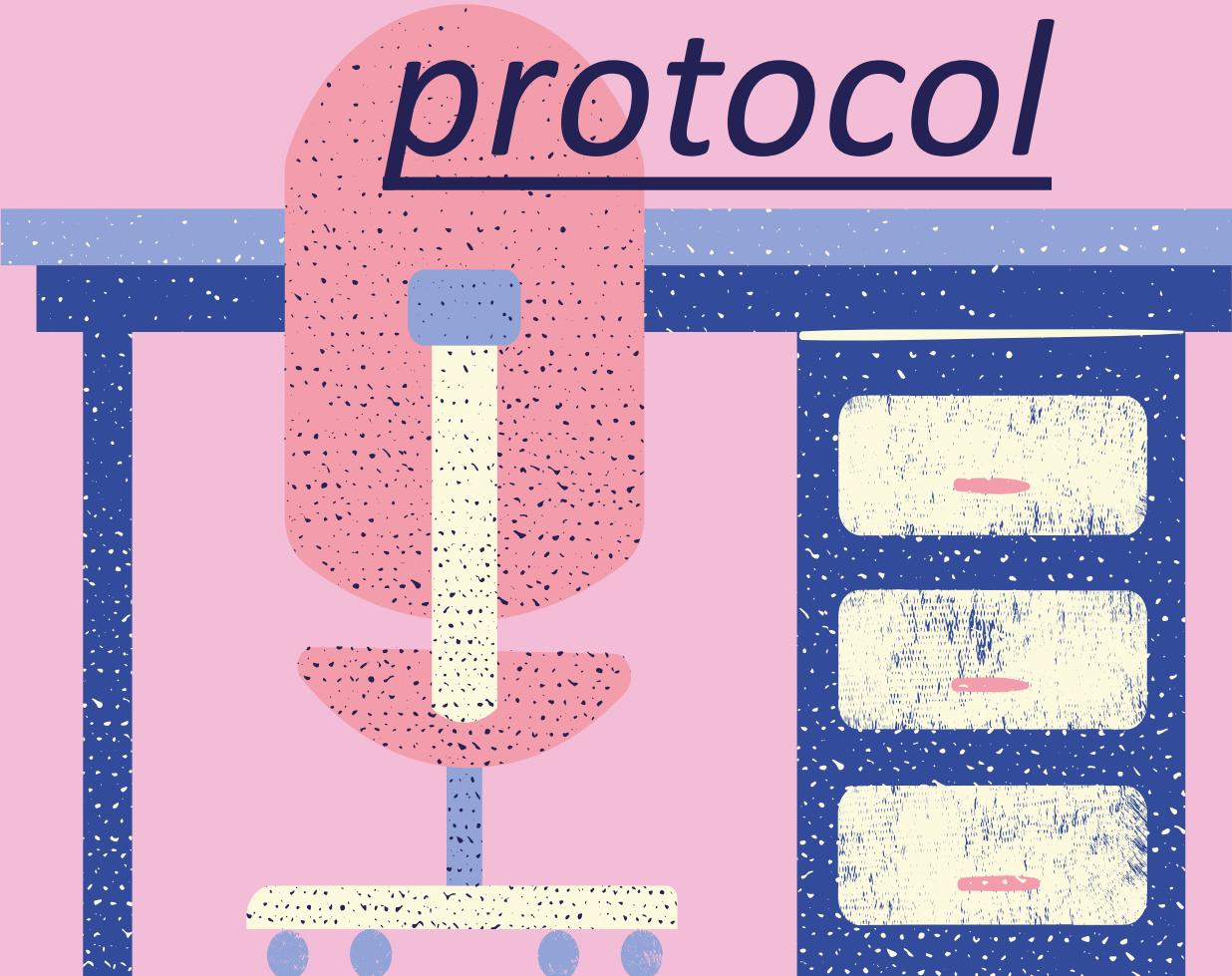
❖ Performance:

FTP can be slower than other file transfer protocols, such as SFTP and FTPS.

FTP is a reliable and versatile protocol that can be used for a variety of file transfer tasks. However, it is important to be aware of its limitations before using it. If security is a top priority, then you should consider using a more secure file transfer protocol, such as SFTP or FTPS.



Recent uses of the File Transfer protocol



E-commerce:

FTP is often used by e-commerce businesses to transfer product images, product descriptions, and other data to their website.

Content distribution:

FTP is often used by content distributors to transfer large media files, such as movies and TV shows, to their servers.

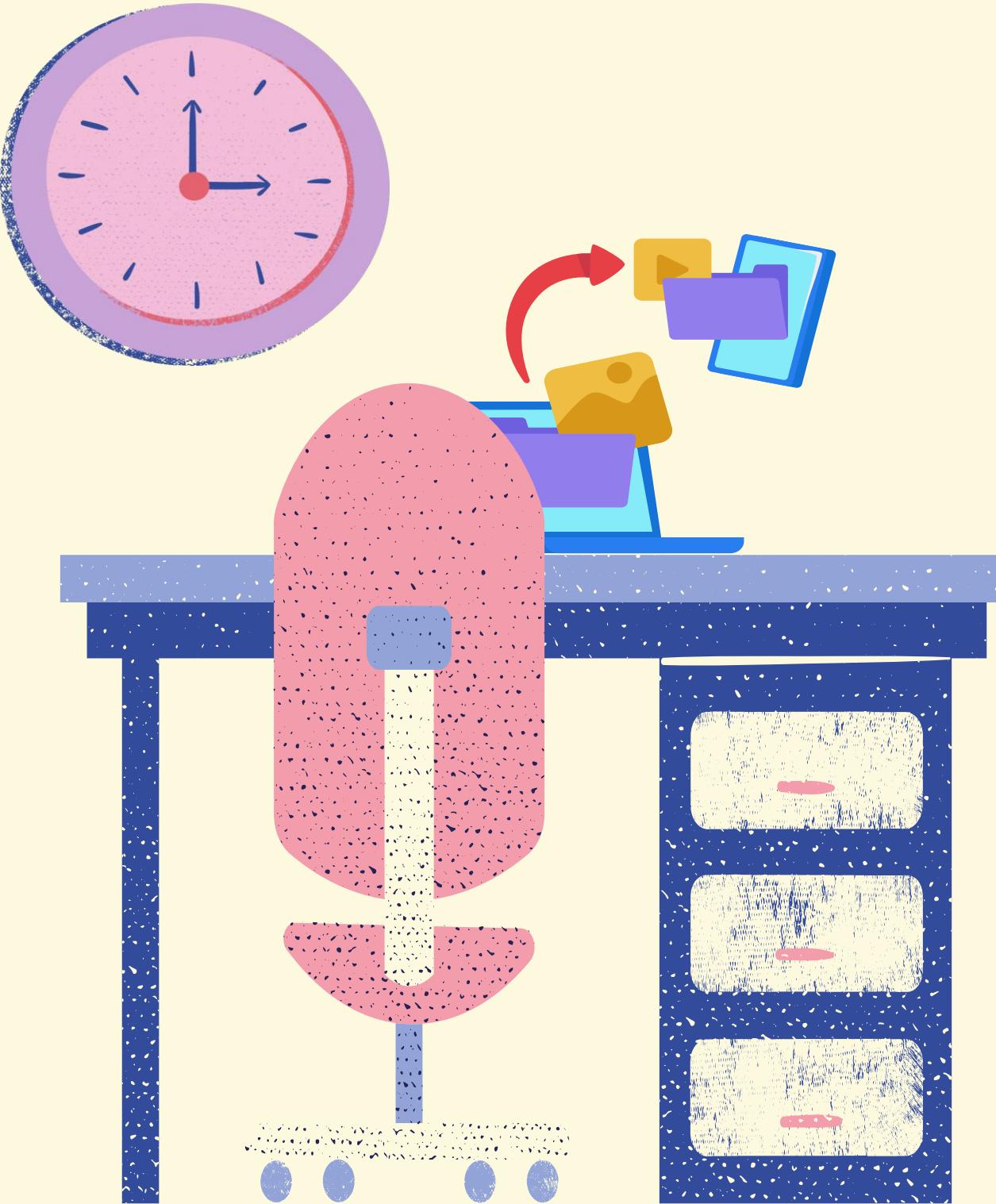
Software development: FTP is often used by software developers to transfer source code and other files between their computers and remote servers.

Backup and recovery:

FTP can be used to back up files to a remote server. This can be a convenient way to back up files, as it does not require any special software or hardware.

File sharing:

FTP can be used to share files with other users. This can be done by setting up an FTP server and giving other users the server's address and login credentials.



Here are some specific examples of recent uses of FTP:

- The Linux Mint distribution uses FTP to download updates from its servers.
- The WordPress content management system uses FTP to upload files to the website.
- Many software companies use FTP to distribute their software to users.
- Some businesses use FTP to transfer customer data between their computers and their servers.

Implementation Of FTP

- The client connects to the server on port 21.
- The client sends a greeting message to the server.
- The server sends a greeting message to the client.
- The client authenticates itself to the server.
- The client requests a file transfer.
- The server transfers the file to the client.
- The client closes the control connection.
- The client closes the data connection.

➤ Code: <https://github.com/manoharreddyvoladri/ftp>

client

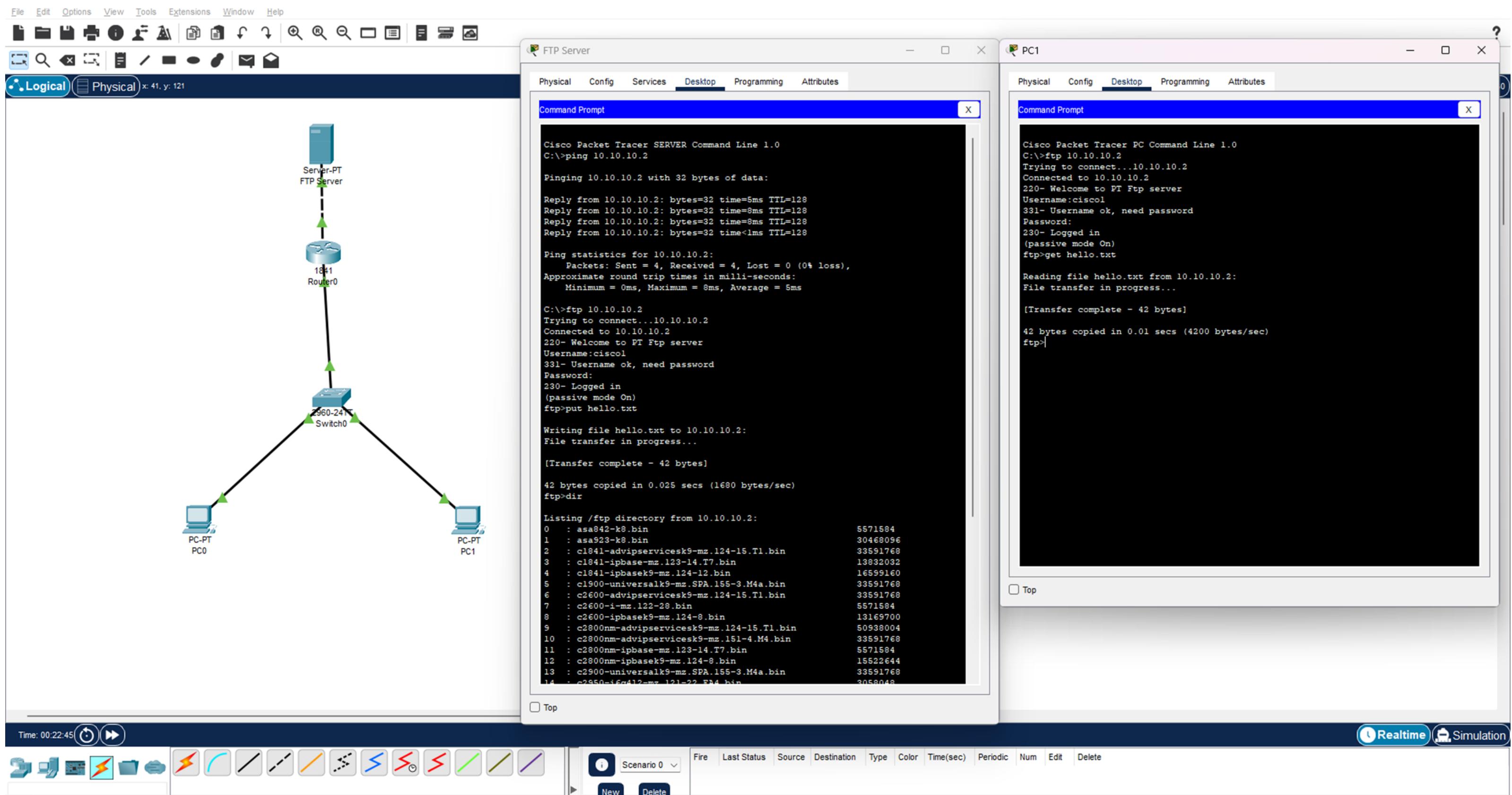
```
(hkdevs@HARISHKALVA) [~/manohar]
$ nvim client.c
(hkdevs@HARISHKALVA) [~/manohar]
$ gcc client.c -o client.o
(hkdevs@HARISHKALVA) [~/manohar]
$ ./client.o
Enter the source file name :
^C
(hkdevs@HARISHKALVA) [~/manohar]
$ touch demofile.txt
(hkdevs@HARISHKALVA) [~/manohar]
$ ./client.o
Enter the source file name :
demofile.txt
(hkdevs@HARISHKALVA) [~/manohar]
$ nvim demofile.txt
(hkdevs@HARISHKALVA) [~/manohar]
$ ./client.o
Enter the source file name :
demofile.txt
My Name is Manohar Reddy
21BRS1177
```

server

```
(hkdevs@HARISHKALVA) [~/manohar]
$ nvim server.c
(hkdevs@HARISHKALVA) [~/manohar]
$ gcc server.c -o server.o
(hkdevs@HARISHKALVA) [~/manohar]
$ ./server.o
Binded
Listening...
Client message
File Name :
Segmentation fault

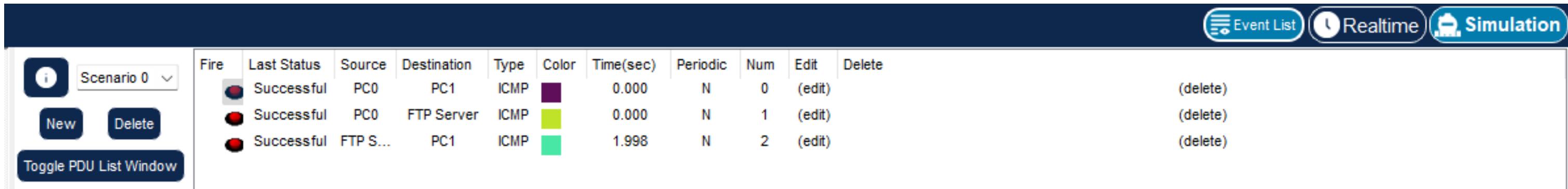
(hkdevs@HARISHKALVA) [~/manohar]
$ ./server.o
Binded
Listening...
Client message
File Name : demofile.txt

File Transferred
```



Execution

Cisco Packet Tracer Student :



Terminal:

```
[hkdevs@HARISHKALVA]~/manohar]$ nvim server.c
[hkdevs@HARISHKALVA]~/manohar]$ gcc server.c -o server.o
[hkdevs@HARISHKALVA]~/manohar]$ ./server.o
Binded
Listening...
Client message
File Name :
Segmentation fault
[hkdevs@HARISHKALVA]~/manohar]$ ./server.o
Binded
Listening...
Client message
File Name : demofile.txt

File Transferred
[hkdevs@HARISHKALVA]~/manohar]$
```

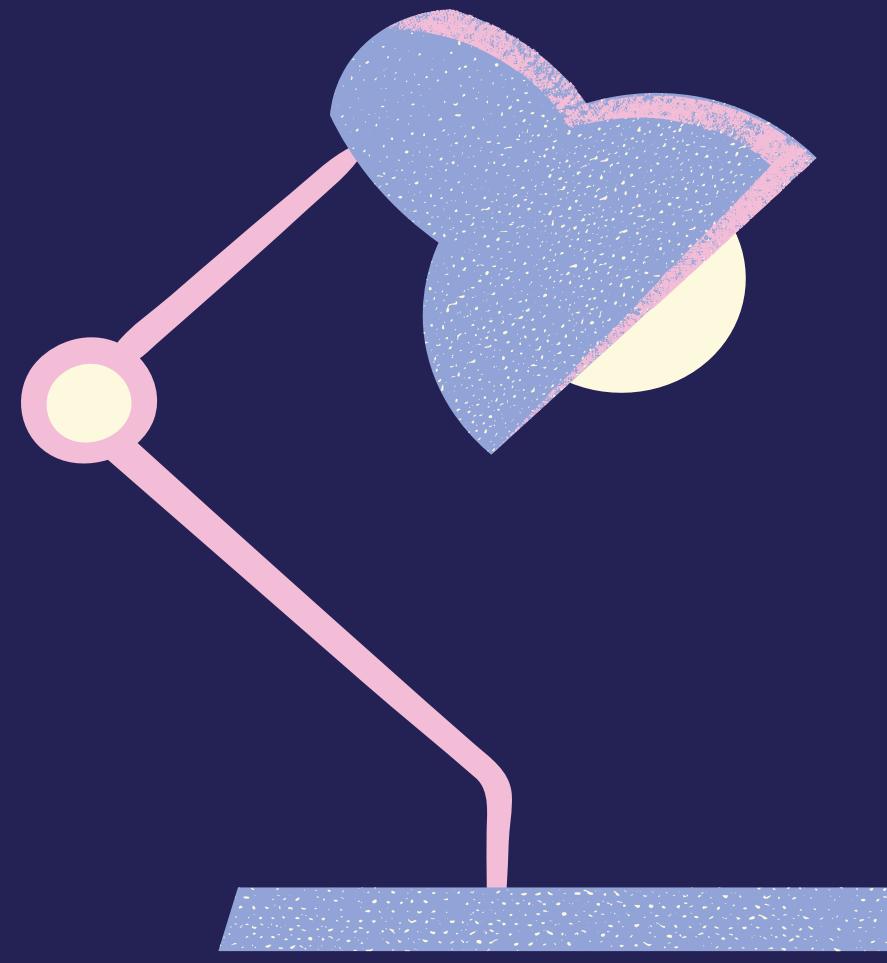
```
[hkdevs@HARISHKALVA]~/manohar]$ nvim client.c
[hkdevs@HARISHKALVA]~/manohar]$ gcc client.c -o client.o
[hkdevs@HARISHKALVA]~/manohar]$ ./client.o
Enter the source file name :
^C
[hkdevs@HARISHKALVA]~/manohar]$ touch demofile.txt
[hkdevs@HARISHKALVA]~/manohar]$ ./client.o
Enter the source file name :
demofile.txt
[hkdevs@HARISHKALVA]~/manohar]$ nvim demofile.txt
[hkdevs@HARISHKALVA]~/manohar]$ ./client.o
Enter the source file name :
demofile.txt
My Name is Manohar Reddy
21BRS1177
```



**Manohar Reddy - 21BRS1177
Varun kumar - 21BRS1540
T.Nivas - 21BRS1311**

Computer Networks

Slot : E2+TE2



Thank you Ma'am.