FTP

- **File Transfer Protocol (FTP)** is the standard protocol provided by TCP/IP for copying a file from one host to another.

- For example, two systems may use different file name conventions. Two systems may have different ways to represent data. Two systems may have different directory structures.

- All of these problems have been solved by FTP. FTP is a better choice to transfer large files or to transfer files using different formats.
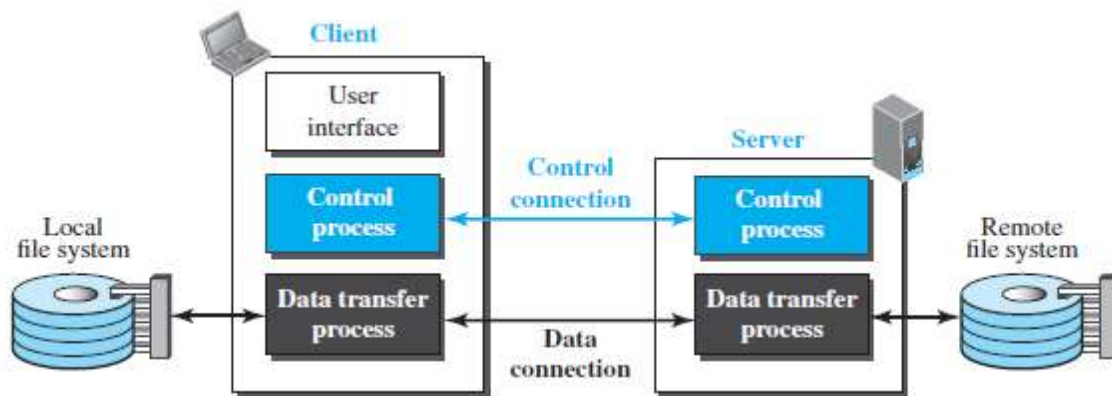

Fig: FTP.

- Figure shows the basic model of FTP.

- The client has three components: the user interface, the client control process, and the client data transfer process. The server has two components: the server control process and the server data transfer process.

- The control connection is made between the control processes. The data connection is made between the data transfer processes.

Two Connections:

- The two connections in FTP have different lifetimes. The control connection remains connected during the entire interactive FTP session.]

- The data connection is opened and then closed for each file transfer activity.

- FTP uses two well-known TCP ports: port 21 is used for the control connection, and port 20 is used for the data connection.

Control Connection:

- For control communication, FTP uses the same approach as TELNET.

- Simple method is adequate for the control connection because we send one command (or response) at a time. Each line is terminated with a two-character (carriage return and line feed) end-of-line token.

- During this control connection, commands are sent from the client to the server and responses are sent from the server to the client.

Command	Arguments	Description
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	Port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed
RNTO	File name (new)	Rename the file
STOR	File name(s)	Store files; file(s) are transferred from client to server
STRU	**F**, **R**, or **P**	Define data organization (**F**: file, **R**: record, or **P**: page)
TYPE	**A**, **E**, **I**	Default file type (**A**: ASCII, **E**: EBCDIC, **I**: image)
USER	User ID	User information
MODE	**S**, **B**, or **C**	Define transmission mode (**S**: stream, **B**: block, or **C**: Compressed

Table: Some FTP commands.

- Every FTP command generates at least one response. A response has two parts: athree-digit number followed by text.

- The numeric part defines the code; the text part defines needed parameters or further explanations.

Code	Description	Code	Description
125	Data connection open	250	Request file action OK
150	File status OK	331	User name OK; password is needed
200	Command OK	425	Cannot open data connection
220	Service ready	450	File action not taken; file not available
221	Service closing	452	Action aborted; insufficient storage
225	Data connection open	500	Syntax error; unrecognized command
226	Closing data connection	501	Syntax error in parameters or arguments
230	User login OK	530	User not logged in

Table: Some response in FTP.

Data Connection:

- The data connection uses the well-known port 20 at the server site. However, the creation of a data connection is different from the control connection. The followingshows the steps:
1. The client, not the server, issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.
2. Using the PORT command the client sends this port number to the server.
3. The server receives the port number and issues an active open using the wellknown port 20 and the received ephemeral port number.

Communication over Data Connection:

- The purpose and implementation of the data connection are different from those of the control connection. We want to transfer files through the data connection.
- The client must define the type of file to be transferred, the structure of the data, and the transmission mode.
- three attributes of communication: file type, data structure, and transmission mode.

File Type:

- FTP can transfer one of the following file types across the data connection: ASCII file, EBCDIC file, or image file.

Data Structure:

- FTP can transfer a file across the data connection using one of the following interpretations of the structure of the data: *file structure, record structure,* or *page structure*.

Transmission Mode

- FTP can transfer a file across the data connection using one of the following threetransmission modes: *stream mode, block mode,* or *compressed mode*.
- The stream modeis the default mode; data are delivered from FTP to TCP as a continuous stream ofbytes. In the block mode, data can be delivered from FTP to TCP in blocks.

File Transfer

- File transfer occurs over the data connection under the control of the commands sent over the control connection. However, we should remember that file transfer in FTP means one of three things: *retrieving a file* (server to client)*, storing a file* (client to server), and *directory listing* (server to client).
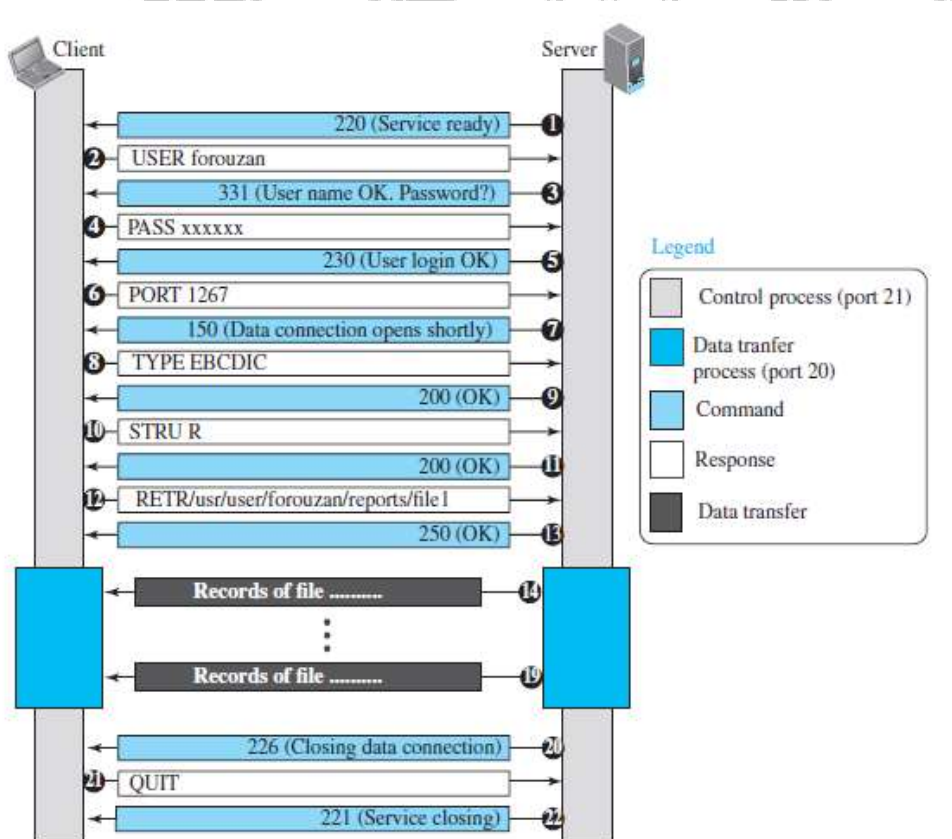
```
$ ftp voyager.deanza.fhda.edu
Connected to voyager.deanza.fhda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and PASS.
Name (voyager.deanza.fhda.edu:forouzan): forouzan
331 Please specify the password.
Password:*********
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
227 Entering Passive Mode (153,18,17,11,238,169)
150 Here comes the directory listing.
drwxr-xr-x    2    3027    411    4096    Sep 24    2002    business
drwxr-xr-x    2    3027    411    4096    Sep 24    2002    personal
drwxr-xr-x    2    3027    411    4096    Sep 24    2002    school
226 Directory send OK.
ftp> quit
221 Goodbye.
```

Fig: Shows an example of using FTP for retrieving a file.

Security for FTP:

- Although FTP requires a password, the password is sent in plaintext (unencrypted), which means it can be intercepted and used by an attacker. The data transfer connection also transfers data in plaintext, which is insecure.

- To be secure, one can add a Secure Socket Layer between the FTP application layer and the TCP layer. In this case FTP is called SSL-FTP.