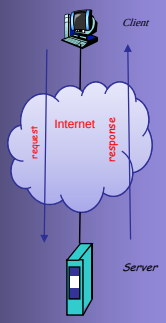


INTERNET PROTOCOLS AND CLIENT-SERVER PROGRAMMING SWE344

Fall Semester 2008-2009 (081)

Module 11: File Transfer Protocol (FTP)

Dr. El-Sayed El-Alfy
Computer Science Department
King Fahd University of Petroleum and Minerals
alfy@kfupm.edu.sa

Objectives

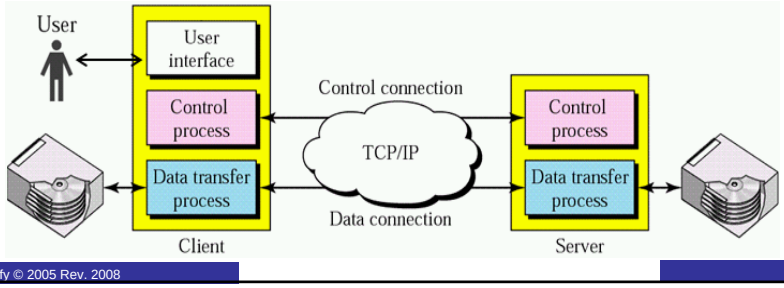
- Learn about the File Transfer Protocol, FTP, [RFC 959]: http://www.ietf.org/rfc/rfc959.txt
  - Its Objectives and Architecture
  - Its functions (commands)
  - Its reply codes

What is FTP?

- FTP is an application layer protocol designed to allow a user on one host to access, and transfer files to and from, another host over a network.
- The design goals of the FTP protocol as stated in RFC 959:
  - To promote sharing of files (computer programs and/or data)
  - To encourage indirect or implicit (via programs) use of remote computers.
  - To shield a user from variations in file storage systems among hosts.
  - To transfer data reliably and efficiently.

Architecture of FTP

- The protocol is designed to have five modules:
  - A client user interface (command or GUI based), that allows a user to interact with the server by issuing FTP commands.
  - A client control process that handles the transmission of commands, reading and interpreting responses.
  - A server control process that reads commands, interprets them and send appropriate responses.
  - Two Data transfer processes (one on ether side) handle the transmission of data resulting from data trasfer commands – e.g. Data resulting from file upload/download commands.

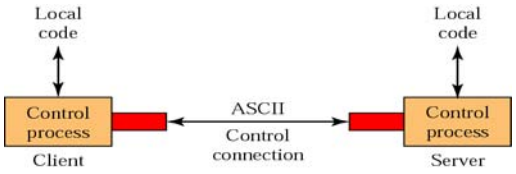


Architecture of FTP …

- FTP establishes two connections between client and server.
  - Control connection (for sending commands and responses)
  - Data connection for data transfer

Control Connection

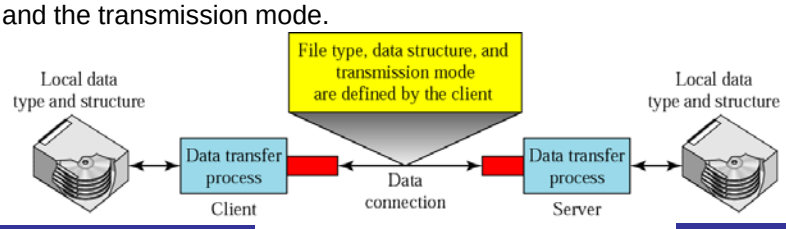
- Uses the default well-known port 21
- Initiated by the client.
- Remains open during the entire FTP session
- Service type used by the IP is minimize delay, because this is an interactive connection between a user (human) and a server.
- Commands and Responses are exchanged in ASCII format.



Architecture of FTP …

Data Connection

- The connection is not permanent -is opened only to service a command and is closed immediately after the data is transferred for the command.
- Uses the default well-known port 20
- A different port number can be negotiated between server and client using PORT or PASV commands. - more on these commands later.
- Data port needs not be on the same host that initiates the control connection. - a user can transfer data between two different servers it has control connections with.
- Client must specify the file type (text or binary), the structure of the data, and the transmission mode.



FTP Commands

- FTP Commands are case insensitive and consist of three to four ASCII characters followed by zero or more parameters.
- Commands can be grouped into three categories:
  - Access Control Commands
  - Transfer Parameter Commands
  - Service Commands

Access Control Commands

- These are commands used to connect to the server and set the working directory.
- NOTE: In the following, commands preceded by * are generally not used in modern computer systems.

FTP Commands ...

Access Control Commands (continued):

Command	Description
USER username	Sends user identification to the server. This is the first command transmitted by the user after establishing a control connection
PASS password (in clear text)	This must immediately follow the USER command, and, for most sites, completes the user's identification. – Note: Many FTP servers allow anonymous read-only access for public documents. For such case, username is anonymous and password is usually an e-mail address.
*ACCT account	Some old sites require account information in addition to username and password.
CWD path	Changes the current working directory to the one specified by path.
CDUP	Changes directory to the parent of the current working directory
*SMNT volume	Loads another data storage structure

FTP Commands ...

Access Control Commands (continued):

Command	Description
REIN	Terminates a USER, flushing all identification and I/O information, except to allow any data transfer in progress to be completed. All parameters are reset to the default settings and the control connection is left open.
QUIT	Terminates a USER and if data transfer is not in progress, the server closes the control connection. If data transfer is in progress, the connection will remain open for result response and the server will then close it.

FTP Commands ...

Transfer Parameter Commands:

- These are commands that are used to set data transfer parameters.
- Most data transfer parameters have default values, and the commands specifying data transfer parameters are required only if the default values are to be changed.
- If a transfer parameter is specified, it remains active until another one is specified.
  - Thus, the server must "remember" the current values.
- The commands may be in any order except that they must precede the FTP service request commands discussed next.

FTP Commands ...

Transfer Parameter Commands (continued):

Command	Description
PORT h1,h2,h3,h4,p1,p2	• Used to specify the IP address and port number to which the server should send data. • h1,h2,h3,h4 represents the IP address in decimal • p1, p2 represents a 16-bit port number. p1 is a decimal representing the left 8 bits and p2 represents the right 8 bits. Note: for a port number, n, p1 = n/256 p2 = n%256

FTP Commands ...

Transfer Parameter Commands (continued):

Command	Description
PASV	This request the server to enter into a passive mode (listen mode) for data connection. The server will respond by sending the IP address and the port number on which it will be listening. The format of the response is: 227 Entering Passive mode (h1,h2,h3,h4,p1,p2) Note: From the above, the port number, n, can be computed as: n = p1*256 + p2

FTP Commands ...

Transfer Parameter Commands (continued):

Command	Description
TYPE A|E |L	Specifies the encoding system of the file being sent The options are: • A for ASCII – default for text files. • E for ABCDIC • I for Image (usually for binary data) – Default format for transferring binary files – File is sent as continuous streams of bits without any interpretation or encoding • L for Local. This must be followed by the number of bytes. Note: Nowadays, most systems use A or I

FTP Commands ...

Transfer Parameter Commands (continued):

Command	Description
*STRU F|R|P	Specifies the structure of the file being sent. Structure allows for check-pointing so that in case of failure, file transfer can resume at the last successful point. The options are: • F for File: contiguous sequence of bytes–default – File has no structure. • R for Record: sequence of records – File is divided into records. – Used only with text files. • P for Page: independent indexed pages – File is divided into pages, with each page having a page number and a page header. Note: Nowadays, most systems use File structure.

FTP Commands ...

Transfer Parameter Commands (continued):

Command	Description
*MODE S|B|C	Specify the transmission mode for the file being sent. The options are: • S for Streams: contiguous sequence of bytes – the default – End-of-file is closing of data connection by sender. – If data is divided into records, each record will have 1-byte end-of-record (EOR) character, and the end of the file will have a 1-byte end-of-file (EOF) character. • B for Block: each block is preceded by a 3-byte header –1st byte is called the block descriptor; next 2 bytes defines the size of block in bytes • C for Compressed: uses run-length compression algorithm. Note: Nowadays, most systems use the Stream mode.

FTP Commands ...

Service Commands:

- The service commands define the file transfer functions that may be requested by a client.
- The parameter is usually the file or directory and the result is normally sent through a data connection.

Command	Description
RETR file	Causes the server-DTP to transfer a copy of the file specified to the client-DTP.
STOR file	Causes the server-DTP to accept the data transferred via the data connection and to store it as a file at the server site. If the file specified exists, its content is overridden.
STOU file	Store unique - same as STOR except that it does not override an existing file
APPE file	Causes the server-DTP to accept the data and store it in a file. If the file specified exists, the data is appended to it
ABOR	Abort the previous FTP service command and any associated transfer of data

FTP Commands ...

Service Commands (continued):

Command	Description
RNFR old_file	Specifies a filename that is to be renamed. RNTO should follow this command.
RNTO new_file	Renames the file specified in RNFR.
DELE file	Deletes the specified file from the server.
RMD directory	Removes the specified directory from the server.
MKD directory	Creates the specified directory on the server.
PWD	Returns the current working directory of the user.
LIST [directory]	Returns the directory listing of the specified directory. - If no directory is specified, the current working directory is listed. - The format of the directory listing is not specified in the protocol. - File attributes, permissions and file size are normally sent. - The listing is returned through a data connection.
NLST [directory]	Similar to LIST except only the list of the file names and directories in returned – no attributes, etc.

FTP Commands ...

Service Commands (continued):

Command	Description
SITE	Returns other services that the server provides. - By using HELP SITE, user can get the list of available arguments for this command. - No arguments are listed by the protocol but some common functions include "QUOTA" to access user quota info. and "CHMOD" to change file permissions.
*STAT	Provides information about data transfer.
HELP [command]	Returns a list of the supported commands by the server. – The optional argument can be used to provide more information about a particular command
SYST	Returns the operating system type on the server.
NOOP	Does nothing
SIZE file	Returns the size of the remote file in bytes
MDTM file	Returns the last-modified time of the file on the remote host in the format "YYYYMMDDhhmmss"

Note: SIZE and MDTM are not standardized, although supported by most servers:

FTP Responses

- Each FTP command is responded with a reply code consisting of 3 digits, followed by an optional text message, followed by CRLF.
- The reply is sent through the control connection.
- A reply message may span more than one line. A dash is placed after the three digits on each message line except the last. A continuation line may also begin with a white space.
- First digit in the reply code has meaning similar to SMTP reply codes

Code	Description
1xy	Positive preliminary reply. Expect another reply before sending another command.
2xy	Positive completion reply. The last command completed successfully.
3xy	Positive intermediate reply. A further command must be sent.
4xy	Transient negative completion reply. The requested action did not take place but can be retried.
5xy	Permanent negative completion reply. The requested action did not take place and should not be retried.

FTP Responses …

- Some typical reply codes:

Code	Description
125	Data connection already opened; transfer starting.
150	File status okay; about to open data connection.
200	Command okay.
211	System status.
212	Directory status.
213	File status
214	Help message.
220	Service ready for new user
221	Service closing control connection.
225	Data connection open; no transfer in progress.
226	Closing data connection
227	Entering Passive Mode (h1,h2,h3,h4,p1,p2).
230	User logged in, proceed.
250	Requested file action okay, completed.
257	"PATHNAME" created.

FTP Responses …

Code	Description
331	User name okay, need password.
332	Need account for login.
350	Requested file action pending further information.
421	Service not available, closing control connection.
425	Can't open data connection.
426	Connection closed; transfer aborted.
450	Requested file action not taken.
451	Requested action aborted: local error in processing.
452	Requested action not taken
500	Syntax error, command unrecognized
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands.
504	Command not implemented for that parameter.
530	Not logged in.
550	Requested action not taken.
551	Requested action aborted: page type unknown.

Sample FTP Session with Telnet

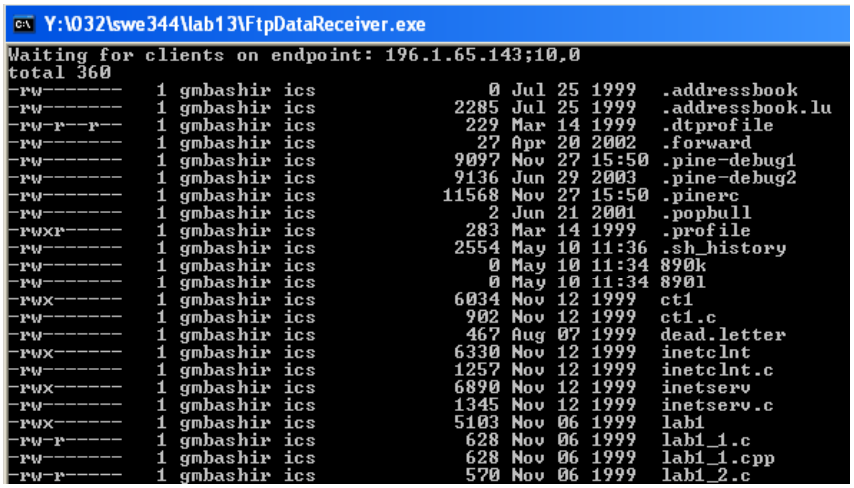
```
Telnet mail.kfupm.edu.sa 21
220 itc6h1 FTP server (Version 4.1 Tue Aug 19 11:38:47 CDT 2003) ready.
user gmbashir
331 Password required for gmbashir.
pass actual_password
230-Last login: Mon May 10 11:34:45 SAUST 2004 on /dev/pts/19 from ics-
    bmghandi.pc.ccse.kfupm.edu.sa
230 User gmbashir logged in.
syst
215 UNIX Type: L8 Version: BSD-44
port 196,1,65,143,10,0
200 PORT command successful.
list
150 Opening data connection for /bin/ls.
226 Transfer complete.
quit
221 Goodbye.

Connection to host lost.
```

Note: For the telnet session to work, there must be another application/socket listening for data connection on the specified end-point.

In this test, we had a console based application **DataReceiver** that does just that.

Sample FTP Session with Telnet …

- The following is the result of the *list* command captured by a **DataReceiver** application running on the specified port:



Resources

- MSDN Library
  - http://msdn.microsoft.com/en-us/default.aspx
- [RFC 959]: http://www.ietf.org/rfc/rfc959.txt
- Books
  - Richard Blum, C# Network Programming. Sybex 2002.
- Lecture notes of previous offerings of SWE344 and ICS343
- Some other web sites and books; check the course website at
  - http://faculty.kfupm.edu.sa/ics/alfy/files/teaching/swe344/index.htm