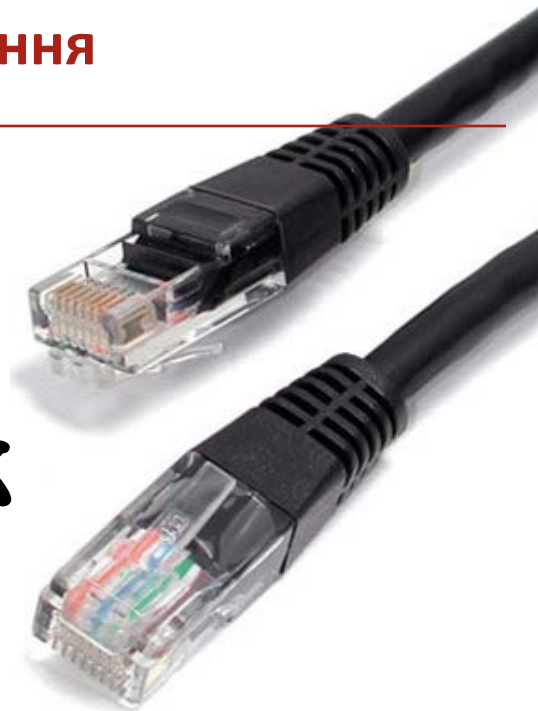


Організація комп'ютерних мереж



к.т.н., ст. викл. Тушницький Р.Б.
ruslan4yk@lp.edu.ua

Лекція 11. Технології прикладного рівня

HTTP

1. Протокол HTTP.
2. Типовий обмін даними за допомогою HTTP.
3. Синтаксис протоколу HTTP. Поняття URL, URI.
4. Типи повідомлень, тіло повідомлень, заголовки повідомлень.
5. Рядок запиту в HTTP. Методи протоколу HTTP.
6. Заголовки HTTP, формат і призначення.
7. Коди відповідей сервера, повідомлення про помилки сервера.

FTP

1. Принципи роботи FTP, FTP сервер та FTP клієнт.
2. Протокол FTP, основні управляючі команди протоколу.
3. Пасивний режим FTP.
4. Поняття анонімного FTP сервера.

Макулатура

1. Оліфер v4: 801-808 + 819-821
2. <http://google.com>

HTTP

HTML-файл = файл, що містить HTML = Hyper Text Markup Language

Протокол HTTP - TCP порт 80

Переваги

1. Простота
2. Розширюваність
3. Поширюваність

Недоліки

1. Великий розмір повідомлень
2. Відсутність «навігації»
3. Відсутність підтримки розподіленості

ПЗ

Клієнт - Google Chrome, Internet Explorer, Mozilla Firefox, Opera, Safari.

Сервер - Apache, Microsoft IIS, lighttpd, nginx. Google Web Server, Resin

Проксі - Squid, UserGate, Multiproxy, Naviscope, Nginx.

Кешування сторінки = proxy server, client, server

URI = URL + URN

URI = Uniform Resource Identifier – уніфікований ідентифікатор ресурсу - символний рядок, що дає змогу ідентифікувати будь ресурс: документ, зображення, файл, службу, ящик електронної пошти і т. д.

URL = Uniform Resource Locator – уніфікований локатор ресурсу - це частина **URI**, яка визначає адресу хоста мережного ресурсу (для немережевих ресурсів ця частина може опускатися).

URN = Uniform Resource Name – уніфікована назва ресурсу - це частина **URI**, яка визначає ім'я ресурсу на хості в локальному просторі імен (в певному контексті).

URI = URL + URN

##

URI => http://ukr.net/w/index.php?title=URI&stable=0#.D0.9E.D1.81.D0.BD.D0.BE.D0

URL => http://ukr.net

URN => /w/index.php?title=URI&stable=0#.D0.9E.D1.81.D0.BD.D0.BE.D0

##

URI => ftp://ftp.dlink.ru/pub/ADSL/

URL => ftp://ftp.dlink.ru

URN => /pub/ADSL/

##

URI => urn:ISBN:0-395-36341-1

URI складається тільки з **URN**, який вказує на ресурс (книгу) 0-395-36341-1 в просторі імен ISBN, передбачається що хост відомий за замовчуванням

URL

URL = Uniform Resource Locator = уніфікований локатор ресурсу - стандартизована адреса певного ресурсу. Включає в себе назву протоколу доступу (HTTP, FTP та ін.) і шлях до ресурсу, формат якого залежить від схеми доступу

<схема>://<логін>:<пароль>@<хост>:<порт>/<URL-шлях>?<параметри>#<якір>

схема - схема запиту до ресурсу, в більшості випадків – мережевий протокол

логін - ім'я користувача, яке використовується для доступу до ресурсу

пароль - пароль вказаного користувача

хост - повністю прописане доменне ім'я хоста в системі DNS або IP-адреса хоста

порт - порт хоста для підключення

URL-шлях - уточнююча інформація про місцезнаходження ресурсу, залежить від протоколу

параметри - рядок запиту з передстрока запроса с переданими на сервер (методом GET) параметрами. Розділювач параметрів – знак **&**.

?параметр_1=значення_1&параметр_2=значення_2

якір - ідентифікатор «якір», який посилається на деяку частину документа

Кодування URL

Якщо ми захочемо використовувати в URL символи кирилиці/ієрогліфи/специфічні символи деякої мови => потрібні нам символи повинні бути перекодовані особливим чином.

Перетворення відбувається в два етапи:

1. Спочатку кожен символ кирилиці кодується в Юнікод (UTF-8) у послідовність з двох байтів
2. Кожен байт цієї послідовності записується в hex представленні

Перед кожним таким hex кодом байта, згідно специфікації URL, ставиться знак відсотка %

Недопустимі символи в URL кодуються по-іншому:

| | | | | | | | | | |
|-----|---|-----|---|-----|---|-----|---|-----|----------|
| %21 | ! | %27 | ' | %3c | < | %5d |] | %7d | } |
| %22 | " | %2a | * | %3d | = | %5e | ^ | %20 | <пробіл> |
| %23 | # | %2c | , | %3e | > | %60 | ` | | |
| %25 | % | %3a | : | %3f | ? | %7b | { | | |
| %26 | & | %3b | ; | %5b | [| %7c | | | |

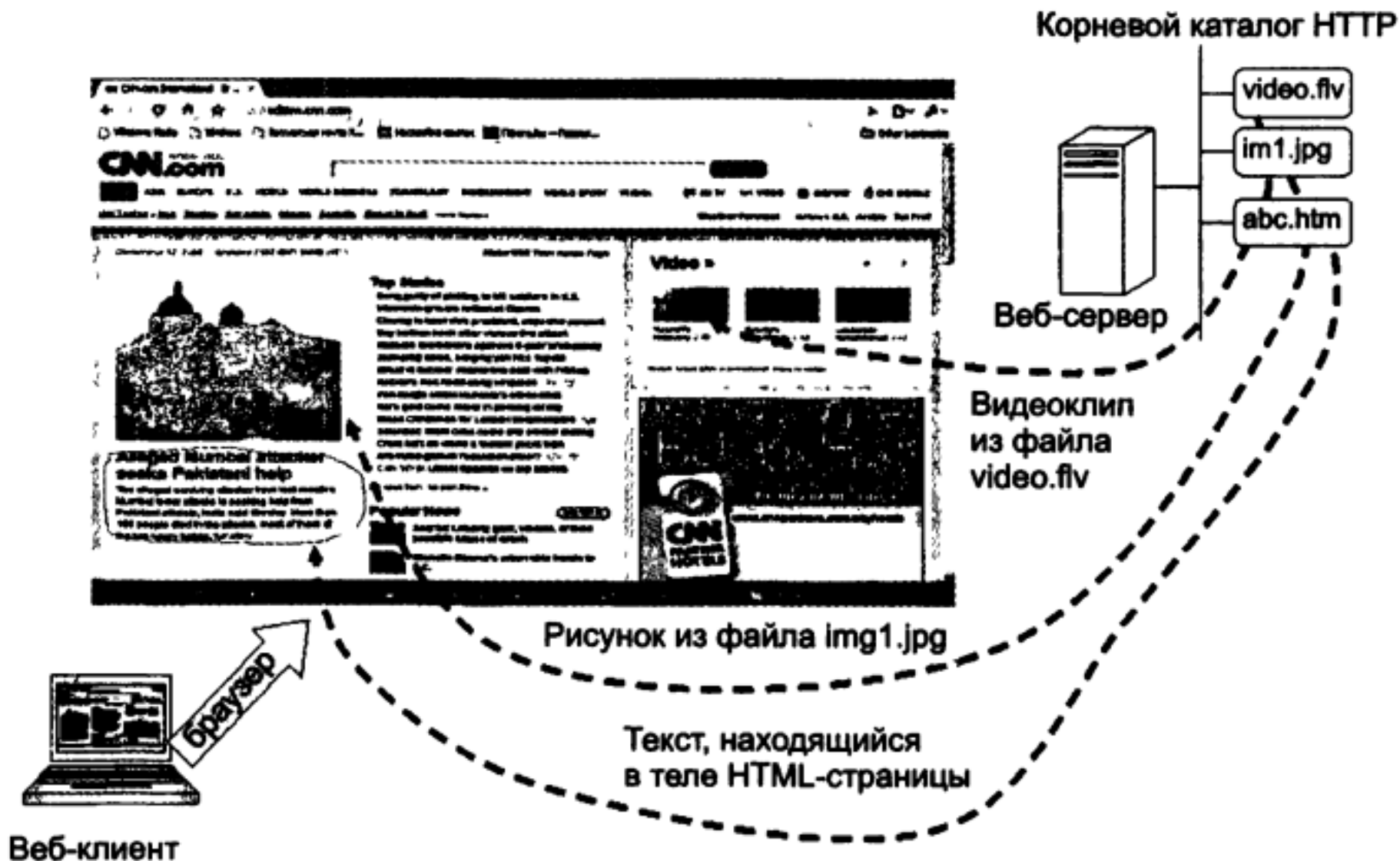
<http://ru.wikipedia.org/wiki/URL>

<http://en.wikipedia.org:80/wiki/Special:Search?search=train&go=Go>

<ftp://myname:mypass@myhost.com:21/etc/motd>

<prospero://myhost.dom//pros/somename>

[file://vms.myhost.edu/disk\\$user/my/notes/note123.txt](file://vms.myhost.edu/disk$user/my/notes/note123.txt)



HTTP

HTTP = Hyper Text Transfer Protocol
v 1.0, 1.1

Обмін повідомленням: запит-відповідь
Тип повідомлення – текстові стандартного формату

Види TCP-з'єднань для транспортування http-повідомлення:

1. Довготривале - передача в 1 tcp-з'єднанні декілька об'єктів.

- i. *послідовна передача запитів з простоями* – новий запит після отримання відповіді
- ii. *конвеєрна передача* – наступний запит **до** отримання відповіді на 1 або декілька попередніх запитів (5-10). http 1.1 .

2. Короткотривале – передача 1 об'єкта.

Таблица 5.1. Форматы стартовых строк и заголовков

| Обобщенная структура сообщения | HTTP-запрос | HTTP-ответ |
|--|--|---|
| Стартовая строка (всегда должна быть первой строкой сообщения; обязательный элемент) | Формат запроса Метод/ URL HTTP/1.x. Пример: GET /books/ books.htm HTTP/1.1 | Формат ответа: HTTP/1.x КодСостояния Фраза. Пример: HTTP/1.0 200 OK |
| Заголовки (следуют в произвольном порядке; могут отсутствовать) | Заголовок о DNS-имени компьютера, на котором расположен веб-сервер. Пример: Host: www.olifer.co.uk | Заголовок о времени отправления данного ответа. Пример: Date: 1 Jan 2009 14:00:30 |
| | Заголовок об используемом браузере. Пример: User-agent: Mozilla/5.0 | Заголовок об используемом веб-сервере. Пример: Server: Apache/1.3.0 (Unix) |
| | Заголовок о предпочтительном языке. Пример: Accept-language: ru | Заголовок о количестве байтов в теле сообщения. Пример: Content-Length: 1234 |
| | Заголовок о режиме соединения. Пример: Connection: close | Заголовок о режиме соединения. Пример: Connection: close |
| Пустая строка | | |
| Тело сообщения (может отсутствовать) | Здесь могут быть расположены ключевые слова для поисковой машины или страницы для передачи на сервер | Здесь может быть расположен текст запрашиваемой страницы |

HTTP: «Метод» «URI» HTTP/«Версія»

OPTIONS - Повертає методи HTTP, які підтримуються сервером.

GET - Запрошує вміст вказаного ресурсу.

Запитаний ресурс може приймати параметри, які передаються в рядку URI:

<http://www.example.net/resource?param1=value1¶m2=value2>.

Згідно зі стандартом HTTP, запити типу GET вважаються ідемпотентними — запити GET повинне приводити до однакових результатів. Це дає змогу кешувати відповіді

HEAD - Аналогічний методу GET, за винятком того, що у відповіді сервера відсутнє тіло.

POST - Передає призначені для користувача дані (наприклад, з HTML-форми) заданому ресурсу. не вважається ідемпотентним

PUT - Завантажує вказаний ресурс на сервер.

DELETE - Видаляє вказаний ресурс.

LINK / UNLINK — встановлює / закриває зв'язок віддаленого ресурсу з іншими.

TRACE - Повертає отриманий запит так, що [клієнт](#) може побачити, що проміжні [сервери](#) додають або змінюють в запиті.

CONNECT - Для використання разом з [проксі-серверами](#), які можуть динамічно перемикається в тунельний режим [SSL](#).

HTTP: відповідь сервера

HTTP/«Версія» «Код статусу» «Опис статусу»

5 класів кодів стану

1xx – informational - інформація про процес передачі

2xx – success - інформація про успішне прийняття і обробки запиту клієнта

3xx – redirection - потрібно здійснити додатковий запит по іншому URL

4xx – client error - помилки на стороні клієнта

5xx – server error - неуспіх при виконанні операцій з вини сервера

200 OK — запит виконаний успішно;

403 Forbidden — доступ до запитаного ресурсу заборонений;

404 Not Found — запитаний ресурс не знайдений.

201 Webpage Created

403 Access allowed only for registered users

507 Insufficient Storage

HTTP: заголовки

Server: Apache/2.2.11 (Win32) PHP/5.3.0
Last-Modified: Sat, 16 Jan 2010 21:16:42 GMT
Content-Type: text/plain; charset=windows-1251
Content-Language: ru

Всі заголовки розділяються на 4 основні групи:

General Headers = основні заголовки – мають включатися в будь-яке повідомлення клієнта і сервера.

Request Headers = заголовки запиту – тільки в запитах клієнта.

Response Headers = заголовки відповіді = тільки для відповіді сервера.

Entity Headers = заголовки сутності – супроводжують кожну сутність повідомлення.

HTTP

GET /wiki/HTTP HTTP/1.1

Host: uk.wikipedia.org

User-Agent: firefox/5.0 (Linux; 5.0.8; en-US; rv:1.8.1.7) Gecko/20070914 Firefox/2.0.0.7

Connection: close

HTTP/1.0 200 OK

Server: Apache

Content-Language: uk

Content-Type: text/html; charset=utf-8

Content-Length: 1234

(далі йде текст запитаної сторінки)

HTTP: перенаправления

GET /about.html HTTP/1.1

Host: example.org

User-Agent: MyLonelyBrowser/5.0

HTTP/1.x **301 Moved Permanently**

Location: <http://example.com/about.html#contacts>

Date: Thu, 19 Feb 2009 11:08:01 GMT

Server: Apache/2.2.4

Content-Type: text/html; charset=windows-1251

Content-Length: 110

(пустий рядок)

<html><body>Click
here</body></html>

HTTP: динамічні веб-сторінки

Мови: php, asp, asp.net, javascript, perl.

Client-side

Server-side

METHOD=GET

<http://www.lp.edu.ua/shedule.php?Department=SoftwareDepartment&Group=PI>.

METHOD=POST

<http://www.lp.edu.ua/shedule.php>

Характеристики:

- метод GET обмежує об'єм переданої інформації;
- метод GET відкрито пересилає введену інформацію;
- сторінку, згенеровану формою з допомогою методу GET – можна додати в закладки

FTP = File Transfer Protocol= Протокол передачі файлів - дає можливість абоненту обмінюватися двійковими і текстовими файлами з будь-яким комп'ютером мережі, що підтримує протокол FTP. Установивши зв'язок з віддаленим комп'ютером, користувач може скопіювати файл з віддаленого комп'ютера на свій, або скопіювати файл з свого комп'ютера на віддалений.

TCP 20 - дані

TCP 21 – команди

Анонімний FTP

Як правило, для роботи по протоколу **FTP** необхідна авторизація за допомогою логіна та пароля. Анонімний **FTP** дає змогу будь-якому бажаючому підключитися до **FTP**-вузла і працювати з ним без авторизації.

FTP: data representation

При передачі даних по мережі, може бути використано 4 подання даних:

ASCII mode = ASCII режим: використовується для тексту. Дані перетворюються, якщо потрібно на "8-бітний ASCII" перед передачею, і (знову ж таки, якщо це необхідно) до представлення символів отримувача. Як наслідок, цей спосіб не підходить для файлів, які містять дані, відмінні від звичайного тексту.

Binary mode = Image mode = Режим зображення = Двійковий режим: машина посилає кожен файл байт за байтом, а одержувач зберігає потік так, як його отримує.

EBCDIC mode = EBCDIC режим: використання для звичайного тексту (**plain text**) між хостами, які використовують набір символів EBCDIC. Цей режим інший за ASCII режим.

Local mode = Локальний режим: Дозволяє двом комп'ютерам з однаковими налаштуваннями здійснювати передачу даних у власному форматі, без необхідності конвертувати його в ASCII.

FTP: data transfer

Передача даних може здійснюватися в будь-якому з трьох режимів:

Stream mode = Поточковий режим: дані передаються у вигляді безперервного потоку, звільнивши FTP від виконання будь-якої обробки. Вся обробка залишилася на ТСР. Непотрібно вказівника кінця файлу, якщо дані діляться на записи.

Block mode = Блочний режим - FTP розбиває дані на декілька блоків (блок заголовка, кількість байт, поле даних), а потім передає його на ТСР.

Compressed mode = Стиснутий режим: дані стискаються за допомогою єдиного алгоритму (як правило, групове кодування).

FTP: Security

Проблеми:

Bounce attacks – атака відмови (сканування портів)

Spoof attacks – фальсифікація даних

Brute force attacks - груба сила

Packet capture (sniffing) - захоплення пакетів

Username protection – захист ім'я користувача

Port stealing – крадіжка порта

FTP не може шифрувати трафік, всі передачі у відкритому вигляді, а також імена користувачів, паролі, команди і дані можуть бути легко прочитані усіма, хто може виконувати захоплення пакетів (sniffing) в мережі.

Ця проблема є спільною для багатьох інтернет-протоколів специфікації (## SMTP, Telnet, POP, IMAP), які були розроблені до створення механізмів шифрування, таких як **SSL** чи **TLS**.

Спільне рішення цієї проблеми полягає у використанні "захищеності", TLS захищених версій небезпечних протоколів (## FTPS для FTP, TelnetS для Telnet і т.д.) або інший, більш безпечний протокол, який може впоратися з таким завданням, наприклад, SFTP/SCP інструменти включені в більшість реалізацій Secure Shell протокол.

FTP: Passive Mode

Спочатку протокол передбачав зустрічне TCP-з'єднання **від сервера до клієнта** для передачі файлу або вмісту каталогу.

Це робило неможливим спілкування з сервером, якщо клієнт перебуває за **IP NAT**, крім того, часто запит з'єднання до клієнта блокується фаєрволом.

Щоб цього уникнути, було розроблено розширення протоколу FTP в пасивному режимі, коли з'єднання для передачі даних теж походить від клієнта до сервера.

Важливим моментом є те, що клієнт встановлює з'єднання з адресою і портом, зазначеним сервером.

Порт сервер вибирає випадковим чином з певного діапазону (49152-65534).

Тому при знаходженні FTP-сервера за NAT, слід явно вказати в налаштуваннях сервера його адресу

FTP: команди

ABOR – перервати передачу файла

CDUP – змінити директорію на вищерозміщену

CWD – змінити диреткорію

DELE – видалити файл (**DELE** filename).

EPSV – увійти в розширений пасивний режим. Застосовується замість PASV.

HELP – Виводить список команд, які приймаються сервером

LIST – повертає список файлів директорії. Список передається через з'єднання даних.

MDTM – повертає час модифікації файла

MKD – створює директорію

NLST – повертає список файлів директорії в більш короткій формі чим **LIST**

NOOP – порожня операція

PASV – увійти в пасивний режим. Сервер поверне адрес і порт, до якого потрібно підключитися, щоб забрати дані. Передача почнеться при введенні таких команд: **RETR**, **LIST** і т.д.

PORT – увійти в активний режим. Наприклад **PORT 12,34,45,56,78,89**. На відміну від пасивного режиму для передачі даних сервер сам підключається до клієнта

PWD – повертає поточну директорію

QUIT – відключитися

REIN – реініціалізувати підключення

FTP: команди

RETR – закатати файл. Перед **RETR** має бути команда **PASV** або **PORT**

RMD – видалити директорію

RNFR і **RNTO** – перейменувати файл: **RNFR** - що перейменувати, **RNTO** - нове ім'я.

SIZE – повертає розмір файла

STOR – закатати файл. Перед **STOR** має бути команда **PASV** або **PORT**

SYST – повертає тип системи (UNIX, WIN, ...)

TYPE – встановлює тип передачі файла (бінарний, текстовий)

USER – ім'я користувача для входу на сервер

PASS – пароль користувача для входу на сервер

`ftp://user001:secretpassword@private.ftp-servers.example.com/mydirectory/myfile.txt`

FTP

220 FTP server ready.

USER ftp //Анонімус

230 Login successful.

PASV

227 Entering Passive Mode (192,168,254,253,233,92) // Клієнт має відкрити з'єднання на переданий IP

LIST

150 Here comes the directory listing. // Сервер передає список файлів в директорії

226 Directory send OK.

CWD incoming

250 Directory successfully changed.

PASV

227 Entering Passive Mode (192,168,254,253,207,56)

STOR gyuyfotry.avi

150 Ok to send data. // Клієнт передає вмістиме файлу

226 File receive OK.

QUIT

221 Goodbye.

FXP = File eXchange Protocol = Протокол обміну файлами - спосіб передачі файлів між двома FTP-серверами безпосередньо, не закачуючи їх на свій комп'ютер. При FXP-сесії клієнт відкриває два FTP-з'єднання до двох різних серверів, запитуючи файл на першому сервері, вказуючи в команді **PORT** IP-адреса другого сервера.

FTP Managers

FileZilla - зручний і простий у користуванні повнофункціональний безкоштовний FTP-менеджер.

WinSCP - вільний графічний клієнт SFTP і SCP, призначений для Microsoft Windows.

Core FTP - потужний безкоштовний клієнт FTP, що містить всі необхідні для обслуговування сайту.

Tunnelier - швидкий і легкий у використанні клієнт SSH що пропонує гнучку підтримку TCP/IP.