

FTP (File Transfer Protocol)

FTP je protokol pro přenos souborů v sítích TCP/IP. Jedná se o netransparentní přenos, netvoří iluzi, že jsou soubory uloženy na lokálním disku jako je tomu u např. sdílení souborů ve Windows. Je využíván například různými internetovými archivy souborů nebo pro upload souborů na webový server.

Protokol sám je starší než TCP/IP. Protokoly TCP/IP získaly svou dnešní podobu zhruba v letech 1977 až 1979, a do sítě ARPANET se začaly prosazovat přibližně od roku 1980, kdy se z ARPANETu postupně stává zárodek dnešního Internetu. První verze protokolu FTP pro netransparentní přenos souborů však pochází již z roku 1971, a od té doby prošla dlouhým vývojem. Od začátku osmdesátých let, kdy i protokol FTP začal využívat transportního protokolu TCP (místo původního transportního protokolu NCP sítě ARPANET) se ale zásadněji nezměnil.

Implementace protokolu FTP vychází z architektury klient-server. Klient je iniciátorem přenosu souborů, server je druhou stranou, která spolupracuje. Klient je typicky představován programem, který si spustíme na počítači, a serverem je démon na počítači, kam nebo odkud chceme soubory přenést. Směr přenosu přitom není důležitý.

Protokol FTP se snaží minimalizovat nároky na různé systémové prostředky - žádá o jejich přidělení až na základě skutečné potřeby, a po jejich použití je zase vrátí. Po celou dobu komunikace mezi klientem a serverem existuje jen to, co zajišťuje vysílání a příjem nejrozličnějších příkazů a odpovědí mezi klientem a serverem, to, co je potřeba pro přenos dat, vzniká dynamicky na základě potřeby.

Řídící spojení

Ta část klienta, která existuje trvale, je tzv. interpret protokolu (PI, Protocol Interpreter). Jeho úkolem je nejprve navázat spojení s interpretem protokolu na serveru, a poté iniciovat jednotlivé akce a řídit jejich průběh. Interpret na serveru čeká na navázání spojení na portu TCP, implicitně 21. Jakmile je spojení navázáno, existuje po celou dobu existence relace, je využíváno pro potřeby řízení relace. Pro komunikaci je použito protokolu Telnet, či je jeho podstatná část implementována, uživateli nic nebrání v tom, aby řídil přenos pomocí terminálového okna.

Příkazy pro řízení přenosu

Interprety protokolu na straně klienta a serveru komunikují prostřednictvím jazyka protokolu FTP. Jazyk je tvořen příkazy klienta a odpověďmi serveru. Příkazy lze rozdělit do tří skupin:

a) Access control commands (příkazy pro řízení přístupu) - umožňují mj. zadat uživatelské jméno a heslo, které pak interpret bude používat k přístupu k souborům.

USER <jméno> přihlášení, uživatelské jméno, poté vyžaduje heslo

PASS <heslo> zadání hesla

QUIT ukončení komunikace

b) Transfer parameter commands (příkazy pro nastavení parametrů přenosu) - umožňují například změnit implicitní čísla portů datového spojení, nastavit režim přenosu, stanovit reprezentaci dat či strukturu přenášeného souboru (viz dále).

PASV - přechod do pasivního módu

PORT <host-port> - adresa a port, kam se má server připojovat v aktivním módu

TYPE <type-code> - typ přenášených dat, může být např. I (image) - binární data, nebo A - ASCII

NOOP - no operation – pravidelným odesláním zabrání klient timeoutu.

ABOR - přerušování datového přenosu

QUIT - ukončení komunikace

c) FTP service commands (výkonné příkazy) - iniciují přenosy souborů a manipulaci s nimi (přejmenovávání, mazání), práci s adresáři (změna adresáře), výpisy obsahu adresáře (vlastní výpis je pak přenášen prostřednictvím datového spojení, lze jej zaslat i pomocí řídicího spojení) apod.

Např.: CWD <pathname> změna pracovního adresáře

PWD zjištění pracovního adresáře

LIST [<pathname>] download výpisu adresáře

RETR <pathname> download souboru ze serveru ke klientovi

STOR <pathname> upload souboru od klienta do serveru

DELE <pathname> smazání souboru

Odpovědi

Server zašle na každý příkaz alespoň jednu odpověď složenou z 3místného desítkového čísla a slovního popisu. Číselný kód je rozhodující, text pouze vysvětluje význam, zobrazuje se uživateli. Číselný kód je trojmístný, první číslice udává výsledek příkazu (propracovanější implementace mohou získat podrobnější informace vyhodnocením druhé a třetí číslice):

1xy předběžná kladná odpověď - požadovaná akce byla úspěšně zahájena, bude následovat další zpráva o jejím průběhu.

2xy kladná odpověď - požadovaná akce byla úspěšně provedena, resp. dokončena.

3xy prozatímní kladná odpověď - příkaz byl přijat, ale jsou nutné ještě další příkazy, např. po použití příkazu USER se vyžaduje heslo.

4xy dočasná záporná odpověď - akce nebyla úspěšná, ale důvod je dočasný, má smysl žádat o nové provedení.

5xy trvalá záporná odpověď - požadovaná akce nebyla úspěšně provedena a nemá smysl se o ni pokoušet znovu.

X a Y jsou druhá a třetí číslice číselného kódu.

Datové spojení

Jakmile se interprety dohodnou, že budou přenášet data, vytvoří si každý svůj přenosový proces (DTP, Data Transfer Process). Tyto procesy si pak mezi sebou naváží samostatné spojení (opět prostřednictvím TCP, tedy spolehlivé a spojované), a jeho prostřednictvím zajistí přenos dat. Navázání řídicího spojení zajišťuje klient, povinnost navázat datové spojení má naopak server (tzv. aktivní mód), nebo klient (tzv. pasivní mód). V aktivním módu sdělí klient serveru, na kterém portu naslouchá (port 20, často také vyšší než 1024) a server vytvoří datové spojení, v pasivním módu klient přikáže serveru naslouchat (server v odpovědi sdělí adresu a port) a sám se připojí.

Pasivní mód je výhodný z hlediska konfigurace firewallů, kdy je jednodušší z hlediska bezpečnosti povolit spojení vytvořené pouze klientem, než předpokládat náhodně vybraný port datového spojení.

Protokol také připouští, aby klient řídil přenos mezi dvěma servery. Řídicí spojení je mezi klientem a oběma servery, datové se vytváří pouze mezi servery. Data tak neproudí přes klientský počítač.

Přenos mezi různými platformami může znamenat problém při interpretaci dat. Každý operační systém má jinak organizovanou strukturu souboru. Protokol FTP se vyrovnává se všemi těmito odlišnostmi v podstatě jediným možným způsobem: zavádí jednotný tvar, ve kterém jsou data skutečně přenášena, a ponechává na obou koncových účastnících, aby zajistili veškeré potřebné převody z/do místních konvencí. Zároveň ale protokol FTP vychází vstříc i případům, kdy by jediný a neměnný přenosový tvar byl neefektivní, a umožňuje jej v určitém rozmezí a po dohodě obou stran měnit. V podstatě lze říci, že tento přenosový tvar má tři stupně volnosti: režim přenosu, strukturu souborů a reprezentaci dat.

Reprezentace dat

Při vlastním přenosu jsou veškerá data přenášena zásadně jako 8-bitové Byty. Protokol FTP však počítá s tím, že počítače, na kterých jsou tato data uchovávána, mohou používat jiné konvence - například 36-bitová slova, jiný způsob ukládání jednotlivých znaků do celých slov, jiný znakový kód než ASCII, apod. Z tohoto důvodu musí být na straně příjemce a odesilatele zajišťovány potřebné konverze. Jejich provádění ovšem vyžaduje, aby obě strany věděly, co mezi sebou přenášají.

ASCII (NVT-ASCII)

Protokol FTP implicitně předpokládá, že data představují ASCII text. Jednotlivé znaky jsou přenášeny jako 8-bitové, a řádky jsou oddělovány dvojicí znaků CR+LF. Odesílatel musí převádět jednotlivé znaky z kódu a způsobu zobrazení, jaký používá jeho počítač, na tento tvar. Příjemce se musí chovat analogicky opačným způsobem.

Protokol FTP však umožňuje, aby se obě strany dohodly na jiném významu přenášených dat. Možnosti jsou následující:

Text v kódu EBCDIC

Data jsou tvořena jednotlivými znaky, tyto jsou opět v 8 bitech, ale pro jejich kódování je místo ASCII použit znakový kód EBCDIC. Varianta vychází vstříc přenosům textů mezi mainframy IBM, které používají tento znakový kód.

Binární data (tzv. image type)

V tomto případě obě strany chápou přenášená data jako spojitou posloupnost bitů, které jsou členěny na 8-bitové Byty jen pro potřeby přenosu.

Byty nestandardní velikosti (tzv. local byte)

Odesílatel může stanovit, jak velké Byty sám používá, a dát příjemci možnost se tomu přizpůsobit. Je si ovšem třeba uvědomit, že jde o „logickou“ velikost Bytu, která slouží oběma stranám jako vodítko pro provádění potřebných konverzí. Nastavení nestandardní velikosti Bytu nemá žádný vliv na skutečnost, že vlastní data jsou doopravdy přenášena v 8-bitových Bytech.

Struktura souborů

Další odlišností, se kterou se protokol FTP musel vyrovnat, je rozdílný pohled na to, jak je soubor vnitřně členěn. Zda je například považován za lineární posloupnost bytů, nebo je členěn na sekvenční záznamy (records), nebo zda je tvořen navzájem nezávislými stránkami (pages), které jsou opatřeny indexy, a mohou vytvářet nespojitý soubor.

Různé systémy mohou pro jeden a tentýž druh souboru používat jinou vnitřní strukturu. Například texty se na většině počítačů uchovávají v textových souborech, tvořených lineární posloupností jednotlivých Bytů, ale například na mainframech IBM jsou textové soubory organizovány jako posloupnosti záznamů pevné délky. I zde pak musí nastoupit vhodná konverze na jedné straně či na obou současně, opět platí zásada, že každý musí vědět, co se vlastně přenáší.

Soubor (File)

Pokud není řečeno jinak, protokol FTP implicitně předpokládá, že přenášený soubor nemá žádnou vnitřní strukturu, a je tvořen lineární posloupností jednotlivých datových Bytů (tato možnost je označována jako file structure).

Záznam (Record)

Soubor členěný na sekvenční záznamy (record structure), např. texty na mainframech IBM.

Stránka (Page)

Soubor členěný na stránky (page structure). Určeno pro soubory s náhodným přístupem (swapovací soubory).

Režim přenosu

Důvodem pro zavedení více režimů byla snaha vyrovnat se s nedokonalostí a omezením přenosových cest. Protokol FTP umožňuje provádět i kompresi dat, a zotavit se z výpadku spojení v průběhu přenosu a po obnovení spojení v něm pokračovat.

Stream mode (proudový mód)

Implicitní režim přenosu je takový, kdy přenášená data jsou chápána jen jako spojitý proud (stream) Bytů, a jako taková jsou také přenášena. Konec souboru reprezentuje ukončení spojení.

Block mode (blokový mód)

Data jsou členěna do bloků, každý z nich je opatřen hlavičkou (ta obsahuje mj. údaj o délce bloku, příznak posledního bloku apod.). Mezi bloky, obsahující vlastní data, lze vkládat i malé speciální bloky (identifikované příznakem ve své hlavičce), tzv. Restart Markers, které ve skutečnosti slouží jako kontrolní body. Po výpadku spojení a jeho následném obnovení je pak možné využít existence těchto značek k indikaci místa, odkud má být přerušovaný přenos opakován. Jinak v bezpečnosti spojení FTP spoléhá jen na protokol TCP.

Compressed mode (komprimovaný mód)

Data jsou přenášena komprimována. Nejde o propracovanou kompresi, řetězec stejných znaků (např. mezer) je pouze zkrácen na jeden tento znak a je přidán počet opakování.

Anonymní FTP

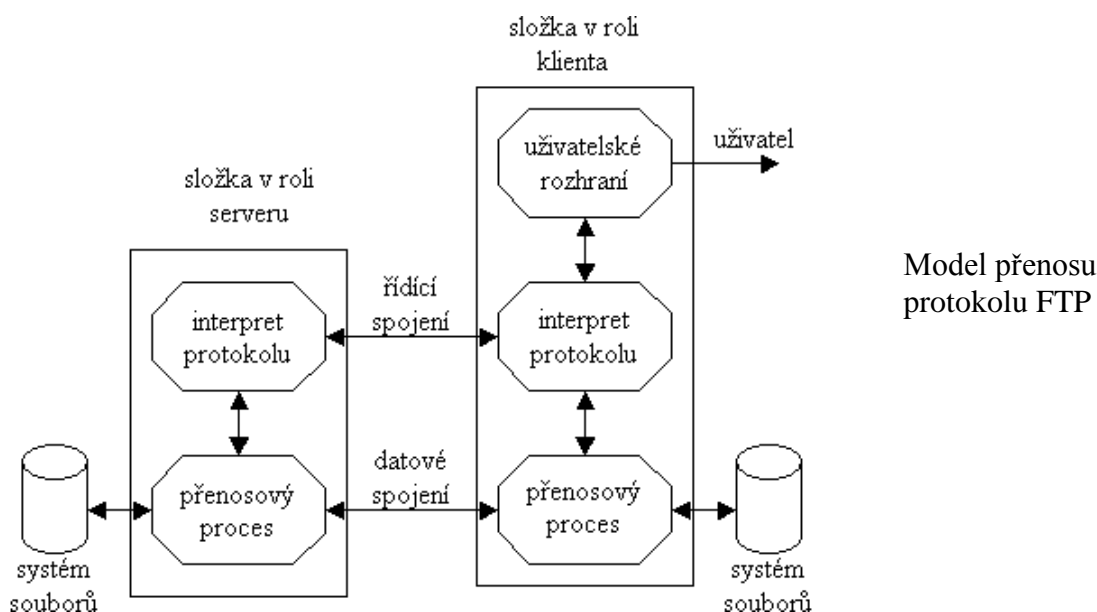
Některé servery (např. archivy dokumentů, freeware, shareware) umožňují tzv. anonymní přístup. Uživatel se přihlásí jako uživatel „ftp“ nebo „anonymous“, případně jinak. Jako heslo uvede většinou svoji emailovou adresu, přičemž většinou nedojde ani k ověření existence emailového účtu, pokud si

server existenci ověří a adresa neexistuje, jednoduše viníka odpojí. Pod anonymním přihlášením nemá většinou uživatel právo zapisovat, pouze číst.

Klientské programy

Pro připojení k FTP serveru slouží veliká škála komerčních i volně šiřitelných programů. Konzolový program Ftp, který je součástí Windows, se ovládá pomocí tzv. uživatelského jazyka, příkazy většinou nejsou totožné s řídicími příkazy protokolu FTP. Kromě specializovaných programů pro přístup k FTP serverům (CuteFTP atd.) jsou FTP klienty součástí souborových manažerů (Total Commander) a HTML editorů (PSPad) a většiny internetových prohlížečů.

Protokol FTP popisuje RFC 959. RFC 2228 popisuje další možnosti protokolu, například šifrování, RFC 1635 je o anonymním FTP a RFC 1579 uvádí řešení problémů s firewalley.



TFTP (Trial FTP)

Protokol TFTP představuje jednodušší variantu protokolu FTP. Používá transportního protokolu UDP (nespolehlivé, nespojované), implicitně na portu 69.

Na rozdíl od FTP používá pouze jedno síťové spojení. Neřeší žádné přihlašování uživatele, přístupová práva. Lze pouze určit, které soubory chceme nabízet ke stažení a server je poskytne každému, kdo si o ně požádá. Protokol neobsahuje funkce pro procházení a výpis adresáře. Uživatel musí znát konkrétní jméno souboru a celou jeho cestu. Soubory přenáší buď jako ASCII text nebo binární data.

Přenos probíhá po blocích o délce 512 B číslovaných od 1. Nultý blok nese informace o souboru. Poslední blok přenosu indikuje jeho velikost, která je menší než 512 B, třeba i 0 B. Klient musí příjem každého bloku potvrdit. Teprve potom může server odeslat další blok. Pokud neobdrží potvrzení do určitého intervalu, opakuje odeslání bloku. Tím se řeší nespolehlivost protokolu UDP. Protokol neumožňuje obnovení přenosu zbytku souboru při výpadku, téměř každá chyba vede k ukončení spojení.

Díky své jednoduchosti lze protokol implementovat do typu PROM a zajistit tak u bezdiskových stanic bootování ze serveru.

Protokol TFTP popisuje dokument RFC 1350, rozšíření o možnosti volby velikosti bloku, délky časového výpadku, omezení velikostí souborů aj. pak dokumenty RFC 1782 až 1785.