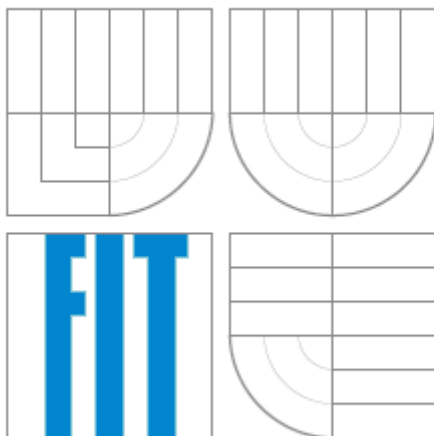


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentace k projektu pro předmět ISA

FTP Klient

20. listopadu 2016

Autor: David Kolečkář, xkolec07@stud.fit.vutbr.cz
Fakulta Informačních technologií
Vysoké Učení Technické v Brně

Obsah

1	Úvod	3
2	FTP protokol	3
2.1	Aktivní a pasivní režim	3
3	Funkce programu	4
4	Ovládání programu.....	5
5	Knihovny	6
6	Závěr	6
A	METRIKY KÓDU	6

1 Úvod

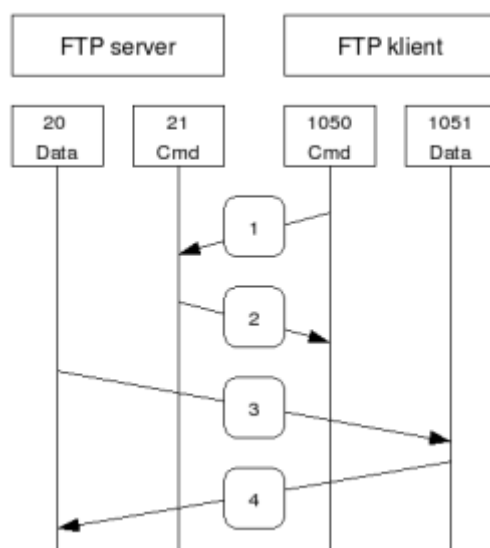
Tento dokument popisuje návrh a implementaci programu v jazyce C++, pro projekt do předmětu síťové aplikace a správa sítí. Úkolem projektu bylo vytvořit FTP klienta. Dokument se skládá z několika částí. V kapitole 2 se věnuji obecnému popisu protokolu FTP. Kapitola 2.1 se zabývá aktivním a pasivním režimem připojení. Následuje kapitola 3, kde rozebírám implementaci jednotlivých funkcí programu. Kapitola 4 je věnována popisu ovládání programu a v kapitole 5 jsou uvedeny použité knihovny.

2 FTP protokol

FTP protokol slouží k přenosu souborů mezi počítači pomocí počítačové sítě. Využívá protokol TCP/IP a je nezávislý na platformě. Používá model klient-server. Server poskytuje data, zatímco klient provádí různé operace. Operace jsou volány pomocí různých příkazů, které jsou definovány v rámci protokolu. FTP protokol využívá port 21 k přenášení příkazů a port 20 slouží k vlastnímu přenosu dat. Přenos dat může být binární nebo textový (ascii).

2.1 Aktivní a pasivní režim

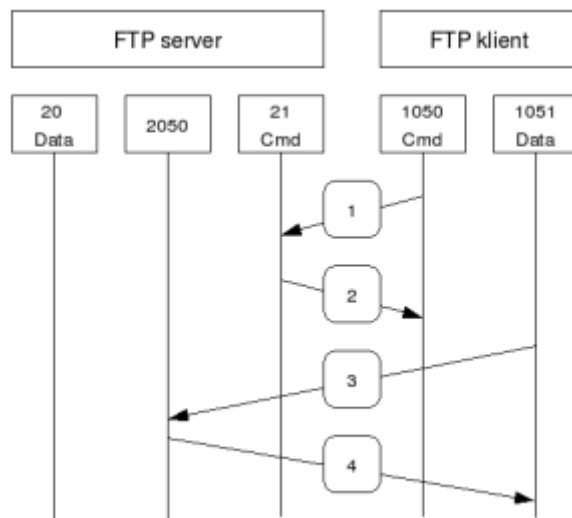
Připojit se k FTP serveru je možné v aktivním nebo pasivním režimu. V aktivním režimu se klient připojuje k serveru na port 21, který je určen pro předávání příkazů. Poté klient pošle příkaz *PORT* s parametry IP adresy a zadaným portem (ve formátu *PORT h1,h2,h3,h4,p1,p2*). Server se poté připojí zpět z portu 20 na port zasláný klientem.



Obrázek 2.1 Ukázka aktivního režimu

V pasivním režimu se klient připojuje k serveru na port 21 pro předávání příkazů. Poté pošle příkaz *PASV*, který zpátky klientovi pošle číslo portu pro datové spojení.

Klient následně iniciuje komunikaci na portu, který dostal. Po tomto kanále si předávají data.



Obrázek 2.2 Ukázka pasivního režimu

3 Funkce programu

K přihlášení na FTP server používám příkaz *USER* pro uživatelské jméno a příkaz *PASS* pro heslo. Heslo i jméno čtu ze zadaného souboru, pokud soubor není v požadovaném formátu, na standartní chybový výstup je vypsána chyba. Pokud není zadán parametr pro pasivní nebo aktivní mód, automaticky používám pasivní mód pro všechny operace.

Pro výpis adresářové struktury celého serveru používám příkaz *LIST*, který vrátí všechny adresáře a soubory zadané složky. Tento příkaz je podobný unixovému příkazu *ls -l*. Následně výstup tohoto příkazu rozebírám na jednotlivé složky a soubory. K tomu mi pomáhá první znak z každého řádku, pokud je to pomlčka jedná se o soubor, jeho cestu a název vypíši na standartní výstup. A pokud je to znak *d* jedná se o adresář, jeho cestu také vypíši. Všechny adresáře si uložím do vektoru adresářů a následně znovu spustím příkaz *LIST* na další adresář ve vektoru. V zadání bylo uvedeno, že se má vypsát kompletní adresářová struktura serveru. Což znamená, že příkaz *LIST* jsem začal provádět s parametrem lomítka. Avšak na testovacím serveru se vypisují i systémové soubory. Proto jsem zvolil implementaci, kde volám příkaz *LIST* s adresářem, ve kterém se nacházím po přihlášení na server.

Pro stažení souboru používám příkaz *RETR*. Aby program mohl stahovat i binární soubor, tak před příkazem *RETR* zavolám příkaz *TYPE I*. Ten umožňuje binární přenos souborů. Pokud je zadán parametr *-P* soubor uložím do zadaného adresáře,

ale jen pokud adresář existuje. Jestliže ne, na standardní chybový výstup je vypsána příslušná chyba.

Pro nahrání souboru na server používám příkaz *STOR*, kterému také předchází příkaz *TYPE I*. Pokud je zadán parametr s cestou kam se má soubor uložit, zavolám příkaz *CWD* s příslušnou cestou. Pokud zadaná cesta neexistuje, je vypsána příslušná chyba.

Pro smazání souboru zavolám příkaz *DELE*. Pokud je zadán parametr s cestou k souboru, zavolám příkaz s příslušnou cestou a názvem souboru. Pokud zadaný soubor neexistuje, je vypsána chyba na standardní chybový výstup.

Pro pasivní režim používám příkaz *PASV*, který mi vrátí adresu serveru a číslo portu, na kterém bude probíhat datový přenos. Při aktivním režimu využívám příkaz *PORT* s parametry IP adresy a čísla portu (ve tvaru *h1,h2,h3,h4,p1,p2*). Pro IP adresu nejprve uvažuji adresu localhostu, pokud se příkaz *PORT* neprovede korektně, pošlu příkaz *PORT* znovu s IP adresou na dalším rozhraní.

Pro ukončení FTP spojení zavolám příkaz *QUIT*.

4 Ovládání programu

Program má dva povinné parametry, kterými jsou název serveru a název souboru obsahující uživatelské jméno a heslo. Pokud jsou zadány jen tyto dva parametry případně aktivní nebo pasivní mód na standardní výstup je vypsána adresářová struktura celého serveru. Parametry spuštění programu jsou následující:

- h** Program vypíše na standardní výstup nápovědu.
- s** Za kterým se uvede IP adresa nebo doménové jméno serveru. Je to povinný parametr.
- c** S názvem souboru, který obsahuje uživatelské jméno a heslo pro přihlášení k FTP serveru. Jedná se taky o povinný parametr.
- a** Číslo portu, kde bude inicializováno datové spojení. Jedná se o aktivní mód komunikace.
- p** Pasivní mód komunikace.
- u** Název souboru, který se uloží na server.
- d** Název souboru, který se stáhne ze serveru.
- r** Název souboru, který se má smazat ze serveru.
- P** Cesta k souboru je to doplňkový přepínač k parametrům **-u -d -r**.

5 Knihovny

Při implementaci programu jsem využil následující knihovny:

<stdlib.h>	<fstream>	<unistd.h>
<stdio.h>	<stdbool.h>	<sstream>
<string.h>	<netdb.h>	<vector>
<iostream>	<arpa/inet.h>	<ifaddrs.h>

6 Závěr

Program je překládán překladačem g++ (s přepínači *-Wall -Wextra -pedantic*), pro překlad slouží soubor *Makefile*. Program byl úspěšně otestován na operačním systému Linux Ubuntu verze 12.04.

A METRIKY KÓDU

Počet souborů: 1 soubor

Počet řádků zdrojového textu: 702 řádků

Velikost statických dat: 16 484B

Velikost spustitelného souboru: 68 353B (systém Linux Ubuntu, 32 bitová architektura, při překladu bez ladících informací)

Reference

[1] J. POSTEL. File transfer protocol (FTP) [online]

<https://www.ietf.org/rfc/rfc959.txt>

[2] O. KREJČÍK. Úvod, trochu teorie a instalace VSFTPD [online]

<http://www.tachec.org/howto/vsftpd/introduction>