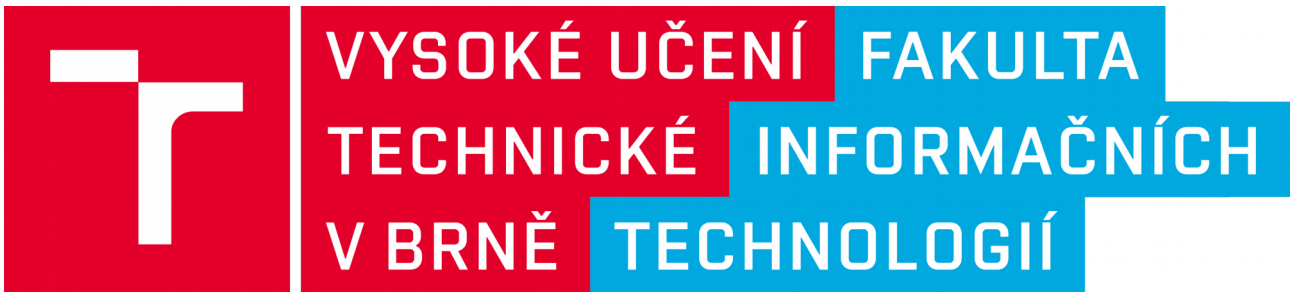
Vysoké učení technické v Brně
Fakulta informačních technologií



Dokumentácia ku projektu ISA

FTP client

Klára Mihalíková <*xmihal05*> 20.11.2016

Obsah

Úvod

Táto správa vznikla ako dokumentácia k projektu do predmetu *Síťové aplikace a správa sítí (ISA)* a zaoberá sa vysvetlením problematiky aplikácie *fclient*, popisuje jej implementáciu a použitie.

1 Dôležité pojmy

1.1 FTP

File transfer protocol, je služba pre prenos súborov po sieti, využívajúca výhradne TCP protokol. FTP je nezávislé na používanom operačnom systéme. Je definovaný štandardom RFC 959.
Služba je neobvyklá v tom, že využíva dva porty(toky), jeden na prenos príkazov(kontrolný port) a druhý na prenos dát. Štandardný kontrolný port serveru je 21. Dátový port závisí na type dátového prenosu, ktorý môže byť aktívny, alebo pasívny. Viac v kapitole 1.1.1 a 1.1.2.

1.1.1 Aktívne FTP

Pri aktívnom FTP sa klient pripája z náhodného neprivilegovaného portu na štandardný kontrolný port serveru, avšak dátový prenos neotvára. Ten otvára server, a to na adrese a porte definovanými klientom cez príkaz `PORT`, na ktorom klient už naslúcha. Server neotvára dátový prenos okamžite, vykoná tak až v prípade, kedy prijme príkaz, kvôli ktorému musí dátový prenos otvoriť (napr. `STOR, RETR, NLST`, atď.).

1.1.2 Pasívne FTP

Pasívne FTP sa od aktívneho líši v tom, že oba toky (ako kontrolný, tak dátový) otvára klient. Do pasívneho módu vstupujeme príkazom `PASV` na kontrolný port serveru, ktorý odpovedá správou obsahujúcou návratový kód `227`(pri nechybovom vstupe do pasívneho režimu) a ip adresu s číslom portu, na ktorom server naslúcha. Ip adresa a číslo portu sú prijaté vo formáte `(i1,i2,i3,i4,p1,p2)`, kde `i1-i4` označujú ip adresu oddelenú čiarkami a `p1` a `p2` číslo portu, ktoré získame výpočtom:

```
dátový port = p1*256 + p2.
```

Príkaz PASV musíme opätovne posielať pred každým dátovým prenosom.

2 Implementácia

Program je implementovaný v jazykoch C/C++. Formálne ide skôr o jazyk C, využívajúci niektoré knižnice jazyka C++. Aplikácia je vytvorená pre operačný systém Linux/Ubuntu.

2.1 Implementačné riešenie jednotlivých častí projektu

2.1.1 Parsovanie prijatých parametrov

K rozparsovaniu prijatých parametrov je využitá funkcia getopt() definovaná knižnicou getopt.h. Vo funkcii getopt() sú definované všetky použiteľné parametre s informáciou o tom, či majú, alebo nemajú obsahovať argument. Prenášaný argument je uložený do položky optarg, odkiaľ je ukladaný do adekvátnych globálnych premenných.

Po úspešnom rozparsovaní prebieha kontrola viacnásobného použitia jednotlivých parametrov, či použitie nekombinovateľných parametrov. Oba prípady ukončia beh programu chybou.

2.1.2 Parsovanie súboru s prihlasovacími údajmi

Parsovanie súboru s prihlasovacími údajmi je realizované v implementovanej funkcii void getLogInf(log_file, line). Súbor je prechádzaný po riadkoch pomocou getline(), z ktorých je vždy oddelených prvých 10 znakov od zvyšku pomocou line.substr(0,10) a line.substr(10). Týchto prvých 10 znakov je ďalej porovnaných pomocou funkcie compare(), či odpovedajú predpísanej štruktúre. Ak compare vráti 0, zvyšok znakov potom obsahuje užívateľské meno a heslo. Program je naimplementovaný tak, aby heslo nemohlo obsahovať iba prázdny reťazec.

Predpísaná štruktúra súboru:

`username:<SP><username><CRLF>`

`password:<SP><password>`

2.1.3 Komunikácia so serverom cez kontrolný kanál

Zasielanie FTP príkazov serveru je implementované pomocou funkcie `send(int sockfd, const void *buf, size_t len, int flags)`. Po odoslaní príkazu sa čaká na odpoveď serveru pomocou funkcie `recv(int sockfd, void *buf, size_t len, int flags)`, ktorá prijatú správu uloží do `bufferu` typu `void *`.
Správy prijaté zo serveru sú parsované v implementovanej funkcii `int recvMsg(int sockfd, void *buf)`.
Funkcia najskôr pretypuje prijatú správu na `string`, aby sa s ňou jednoduchšie pracovalo. Potom sú oddelené prvé 3 znaky pomocou funkcie `substr(0,3)`, ktoré sú pretypované do `int`, a zasielané ako návratová hodnota funkcie.

V prípade, že je táto hodnota rovná `227`(vstup do pasívneho režimu), je uložená ešte adresa a port. Oddelenie tejto časti prebieha pomocou `substr(26)`, keďže vždy ide o posledné znaky začínajúce na pozícii 27, `sscanf()` pre uloženie číselných hodnôt do samostatných premenných (`ip1-ip4` pre ip adresu a `port1` a `port2` pre výpočet čísla portu pre dátovú komunikáciu).

Výpočet čísla portu: `port1*256+port2`.

Adresa je uložená v správnom formáte s bodkami medzi jednotlivými časťami pomocou `sprintf()`.

2.1.3.1 Mazanie súboru

Keďže pre mazanie súboru zo serveru nepotrebujem mať otvorený dátový prenos, ešte pred zasielaním `PASV` pri pasívnom móde, alebo `PORT a1,a2,a3,a4,p1,p2` pošlem príkaz `DELE` s menom súboru na kontrolný kanál. Pokiaľ mi funkcia `recvMsg(…)` vráti `250` (zmazanie prebehlo úspešne), uvoľním alokovanú pamäť a program ukončím s návratovým kódom `0`, úspech.

2.1.4 Dátový prenos

2.1.4.1 Zaslanie súboru

Súbory sú serveru zasielané pomocou funkcie `sendfile(int *out_fd*, int *in_fd*, off_t **offset*, size_t *count*)`.

2.1.4.2 Prijímanie súboru a výpis adresárovej štruktúry

Súbory sú prijímané funkciou recv(...), rovnakou, akou sú prijímané správy od serveru na komunikačnom kanále. Prijímané informácie sa po bitoch zapisujú do vytvoreného súboru.

Pre výpis adresárovej štruktúry bola myšlienka takáto - prijmem do vytvoreného súboru pre zápis informácie pomocou `recv(...)`, tie pretypujem a pomocou rekurzivnej funkcie `printDir(...)` ich po riadkoch vypisujem na štandardný výstup, pričom pokiaľ ide o adresár, vnorujem sa nižšie a rekurzívne volám funkciu.
Bohužial sa nestihla naimplementovať, a tak bola funkcia odstránená a miesto celkovej adresárovej štruktúri program vypisuje iba obsah koreňového adresára.

2.1.5 Aktívny režim

Ako bolo vyššie spomenuté, pri aktívnom režime vytvára dátové spojenie server. Klient však zasiela serveru správu `PORT`, v ktorej oznamuje, na akej ip adrese a ktorom porte serveru naslúcha. Program počíta s dvomi prípadmi ktoré môžu nastať:

1. klient má priradenú adresu iba na `local loopbacku`
2. klient má priradenú adresu na jednom ďaľšom rozhraní

V prvom prípade netreba zisťovať lokálnu ip adresu, použije sa `127.0.0.1`, ku ktorej sa pridá port predaný argumentom v parametre `-a`.

V druhom prípade program zistuje lokálnu adresu pomocou `getifaddrs()`, ku ktorej sa pridá port predaný argumentom v parametre `-a`.

Po vyriešení tohto problému zasiela klient serveru správu `PORT` a pripravuje sa na dátový prenos. Ten však už nestihol byť naimplementovaný.

2.2 Použité knižnice

```
#include <cstdlib>
#include <cstdio>
#include <iostream>
#include <string.h>
#include <string>
#include <fstream>
#include <getopt.h>
#include <fcntl.h>
#include <sstream>
#include <algorithm>
#include <pthread.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <sys/sendfile.h>
#include <sys/stat.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <ifaddrs.h>
```

2.3 Použitie aplikácie

Spustenie programu:

./fclient [-h/--help] -s 192.168.1.102 -c credentials.txt [-p]|[-a port_number] [-d|-u|-r filename] [-P path]

Vysvetlenie použitia jednotlivých parametrov:

`-h` Spustenie programu s týmto nepovinným parametrom bez argumentu vypíše nápovedu k používaniu programu. Parameter je nekombinovateľný, pri kombinácii s iným parametrom sa beh programu ukončí chybou.

`-s` Parameter `-s` je jedným z dvoch povinných parametrov. Obsahuje argument, ktorý nesie IP adresu, alebo doménové meno FTP serveru

`-c` Druhý povinný parameter je `-c`. Jeho argument odkazuje na textový súbor, obsahujúci prihlasovacie údaje klienta. Tento súbor má pevnú štruktúru uvedenú v kapitole 2.1.2.

`-a` Parameter `-a` určuje, že komunikácia bude prebiehať v aktívnom režime.

Argument `-a` určuje port, na ktorom bude dátová komunikácia prebiehať

`-p` Určuje, že dátová komunikácia bude prebiehať v pasívnom režime

V prípade, že žiaden z parametrov `-a/-p` nebol použitý, program pracuje v pasívnom režime. Tieto parametre nie je možné kombinovať.

`-d` Nesie názov súboru, ktorý bude stiahnutý zo servera. V prípade, že je skombinovaný s parametrom `-P`, bude tento súbor uložený do zložky danej argumentom parametru `-P`.

`-u` Nesie názov súboru, ktorý bude nahraný na server. V prípade, že je skombinovaný s parametrom `-P`, bude tento súbor uložený do zložky danej argumentom parametru `-P`.

`-r` Nesie názov súboru, ktorý bude zmazaný zo servera. V prípade, že je skombinovaný s parametrom `-P`, bude tento súbor zmazaný zo zložky danej argumentom parametru `-P`.

Parametre `-d/-u/-r` sú navzájom nekombinovateľné. Pokiaľ budú použité súčastne, program sa ukončí chybou.

`-P` Argument parametru `-P` obsahuje cestu k súboru určeného parametrami

`-d/-u/-r` a je možné ho použiť iba v kombinácii s niektorým z nich.

Všetky parametre programu je možné použiť iba raz, viacnásobné uvedenie ukončí program chybou.

2.4 Výstup aplikácie

- Pri spustení programu s parametrom `-h,` je výstupom programu nápoveda, ktorá sa vytlačí na štandardný výstup.
- Pri spustení aplikácie iba s povinnými parametrami je výstupom programu adresárová štruktúra koreňového priečinka.
- V prípade, že aplikácia prebehne bez chyby, program vracia hodnotu `0.` Okrem vyššie uvedených prípadov nič iné na štandardný výstup nevypisuje.
- V prípade že aplikácia skončí s chybou, je chybová správa vypísaná na štandardný chybový výstup a program vracia hodnotu `1.`

3 Záver

Program nestihol byť doimplementovaný. Pri aktívnom režime nefunguje nahranie a stiahnutie súboru a výpis adresárovej štruktúry. Pri pasívnom móde nie je dokončený výpis adresárovej štruktúry.

Zdrojový kód by bolo dobré optimalizovať, obmedziť používanie globálnych premenných, zbytočne neduplikovať rovnaké funkcie a lepšie štrukturovať, ale kvoli časovej tiesni pred odovzdaním projektu sa tomu nevenovala dostatočná pozornosť.

3.1 Metriky kódu

- **Počet preložiteľných súborov:** 2 + Makefile
- **Počet riadkov kódu:** 854 (aj s komentármi)
- **Veľkosť spustiteľného súboru:** 49294 B

4 Použité zdroje

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

https://linux.die.net/man/3/getifaddrs

http://man7.org/linux/man-pages/man2/sendfile.2.html

https://linux.die.net/man/2/send

http://man7.org/linux/man-pages/man2/recv.2.html

http://slacksite.com/other/ftp.html