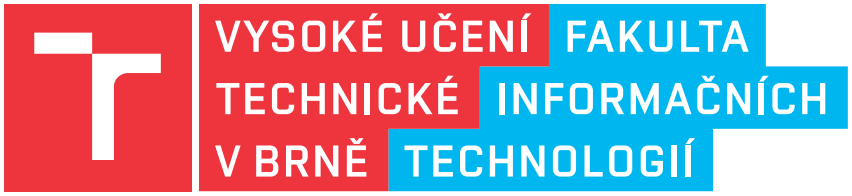

Síťové aplikace a správa sítí

Projektová dokumentace

TFTP Klient

15. listopadu 2021

Marek Sechra (xsechr00)

Obsah

1 Úvod

Tato dokumentace popisuje implementaci TFTP klienta

1.1 Účel programu

Cílem projektu bylo vytvořit program v jazyce C,C++ a nastudování si TFTP protokol a klíčových standardů, které pojednávájí o této problematice. Komunikace TFTP se skládá z dvou stran (klient, server). Implementace a problematika programu se týka klienta.

1.2 Použití programu

Po překladu je možné tento program spustit a pomocí zadaných parametrů nastavit chování klienta, zapís, přenos souboru, cestu, ip adresu serveru apod.

Příklad sputštění programu

```
./mytftpclient
-R -d /home/file.txt -a ::1,69 -c binary
```

Ukázka pomocné hlášky

```
Example of usage: -R/ -d /home/user/readme.txt
```

1.3 Kompilace

Program lze zkompilovat pomocí přiloženého souboru Makefile pomocí příkazu `make` Pro vyčištění adresáře programu lze použít `make clean` Program využívá překladač GCC pro verzi jazyka C11++

1.4 Parametry

Zde jsou popsány implementované parametry.

- parametr -R | -W , povinný parametr, který znamená zápis či čtení souboru odvozeno z (read, write).
- parametr -d , povinný parametr, který popisuje absolutní cestu k souboru na serveru nebo klientovi záležíí na parametru R/W.
- parametr -s nepovinný parametr, který nastavuje velikost dat přenášených v jednom packetu pokud tento parametr není zadán je standartní velikost dat 512B maximální velikost se odvozuje od dané architektury, maximální průchozí hodnota je 1468B jelikož maximální velikost MTU je 1500MB, ale zde je důležité brát ohled na přiložené hlavičky FTP,IP,UDP, které zabírájí 32B.
- parametr -c nepovinný parametr, tento parametr rozhoduje o módu přenosu dat. je použito binary,octet pro základní přenos po 8bitech a ascii čili netascii. bez zadání tohoto parametru se nastaví transfer octet.
- parametr -a, nepovinný parametr, který popisuje ip adresu a port ve formátu

  ```
  IPV4 0.0.0.0,port
  IPV6 ::1,port
  ```

2 Základní pojmy

TFTP, IP, OPCODE, UDP, mód, packet

2.1 TFTP

Protokol TFTP (Trivial File Transfer Protokol) využívá se především pro jednoduchý přenos souborů, obsahuje jen základní funkce TFP. Tento protokol používá port s číslím 69. Využívá transportní protokol UDP, který nezaručuje spolehlivý přenost. Packet je balíček dat nesoucí informaci s ohledem na standarty TFTP. TFTP protokol je možno provozovat na Ipv4 a ipv6. Komunikace je klient - server, kdy klient vždy začiná tuto komunikaci. První odeslány packet je vždycky dotaz na server nesoucí informací o žádosti zápisu nebo opisu konkrétního souboru.

2.2 IP

Program podporuje ipv4 a současně ipv6.

2.3 OPCODE

OPCODE se používá v protokolu TFTP pro značení typu packetu. Je zde problém architektur a kodování na bitové úrovni čili big endian nebo little endian. Typy OPCODE:

- RRQ - 1 . Používá se v prvním dotazu pro značení požadavku o čtení souboru ze serveru.
- WRQ - 2 Používá se v prvním dotazu pro značení požadavku o zápis na server
- DATA - 3 - Značí, že jsou přenášeny data.
- ACK - 4 - Potvrzení doručení správného packetu.
- ERROR - 5 - Chyba během přenosu.
- OACK - 6 - souvísí s parametrem -s pokud je transfer větší než po 512, vrátí se jako potvrzení protistrany s navrhovanou délkou dat v přenosu.

3 Implementační detaily

Tato kapitola popisuje jakým způsobem je program strukturován a jakým způsobem je řešen parametr -s.

3.1 Struktura

Program obsahuje 2 hlavičkové soubory a 2 soubory primární.
argparse.h argparse.c tento systém souborů se věnuje problematice vstupních parametrů a jejich kontrolu správnosti a sounaležitosti. Parsování argumetů probíhá v cyklu pomocí funkce `getopt()`
Je zde struktura `flags` , která je plněna daty pro další práci v klientovi.
Obsahuje

- `RWflag` , což je číselná hodnota, která značí použití přepínače R|W nebo nezadání ani jednoho parametru.
- `multiflag` značka pro požadavek multicastu - dále se nepoužívá, ale byl ponechán z důvodu možnosti nadstavby multicastu v projektu.
- `counterRW` pomocná značka, která když není 0, tak došlo k zadání R,W současně.
- `path` Obsahuje absolutní cestu k souboru.
- `timeout` pro požadavek timeoout - dále se nepoužívá, ale byl ponechán z důvodu možnosti nadstavby multicastu v projektu.
- `highpass` Hodnota značicí hodnotu, která odpovídá velikost dat použitý v přenosu. Datový typ je int.
- `shighpass` Tato hodnota obsahuje stejnou informaci jako předchozí parametr, ale jeho datový typ je řetězec znaků.
- `ipadress` Hodnota ip adresy je implicitně nastavena na localhost v ipv4.
- `ipv6flag` Hodnota, která předává informaci o používání ipv6 v programu.
- `port` číslo portu.

Hlavním modulem je mytftpclient.cpp a jeho hlavičkový soubor s příponou .h

3.2 Vytvoření TFTP hlavičky

Na začátku ve funkce main jsou incializované všechny potřebné struktury pro zahájení internetové komunikace skrz sockety. Po té dochází k vytvoření bufferu a ukazatele. Pro, které se vytvoří potřebná TFTP hlavička tedy hlavička obsahujicí RRQ/WRQ, cestu k souboru a absolutní cestu k souboru včetně modu. Pokud byl zadán parametr -s nastaví i nová hodnota pro velikost datového přenosu.

3.3 Čtení ze souboru

v souboru mytftpclient.c je volána funkce `readtftp()` Kde se vytvoří nový soubor pro zápis dat, které budou přicházet ze serveru. Poté se zahájí komunikace se serverem a a příchozí data jsou zapisovány do souboru. Funkce obsahuje řešení pro případ přichozích dat v ascii kodování i pro posílání OACK, kde je potřeba se dohodnout na velikosti posílaných dat.

3.4 Zápis souboru

Zápis souboru je druhá větev jedné z primárních podmínek v mytftpclient.c . Zde se otevírá soubor a data jsou posílány na server. Problém nastává, při posílání posledního packetu, kde je velikost jiná než bylo dohodnuté se serverem. Je potřeba alokovat nový buffer a naplnit ho korektně poslednímy daty. Zbylou velikost zjištuji pomocí matematické operace modulo a vytvářím počitadlo, které hlídá čas posledního packetu.

3.5 -s blocksize option

v funkci argparse si hlidám jestli hodnota zadaného aprametru nepřekročí požadovanou hodnotu. Dále zvětšuji velikost definovaných bufferu.

Reference

[1] Nový řádek, 2001.

[2] A. Thomas Emberson. TFTP Multicast Option. RFC 2090, February 1997.

[3] Art Harkin and Gary S. Malkin. TFTP Option Negotiation Analysis. RFC 1785, March 1995.

[4] Gary S. Malkin and Art Harkin. TFTP Blocksize Option. RFC 2348, May 1998.

[5] Gary S. Malkin and Art Harkin. TFTP Option Extension. RFC 2347, May 1998.

[6] Gary S. Malkin and Art Harkin. TFTP Timeout Interval and Transfer Size Options. RFC 2349, May 1998.

[7] Dr. Karen R. Sollins. The TFTP Protocol (Revision 2). RFC 1350, July 1992.