

Vysoké učení technické v Brně
Fakulta informačních technologií

Síťové aplikace a správa sítí
TFTP Klient

1 Úvod

Úkolem projektu je implementace klientské části TFTP verze 2 protokolu. TFTP (*Trivial File Transfer Protocol*) je jednoduchý protokol pro přenos souborů nad UDP/IP. Obsahuje pouze dvě operace - čtení souboru ze serveru a zápis souboru na server. Verze 2 tohoto protokolu obsahuje rozšíření popsané níže.

Hlavním zdrojem informací při implementaci projektu byly [RFC 1350](#) a řada RFC dokumentů popisujících způsob fungování jednotlivých rozšíření. Projekt je implementován v jazyce C.

2 Implementace

2.1 Tok programu

TFTP klient má vlastní příkazový řádek (zadávací prompt). Program běží po celou dobu a čeká, až uživatel zadá příkaz. Uvnitř funkce *tftp_client()* je implementován cyklus, kde se při každé iteraci zpracuje příkaz voláním funkce *tftp_process_command()*. Uživatel může program ukončit zadáním prázdného řádku.

2.2 Komunikace

Komunikace probíhá prostřednictvím socketů. BSD UNIX Socket Library poskytuje sadu hlaviček, které definují funkce a struktury umožňující komunikaci na relaci klient-server. Implicitně server běží na adrese 127.0.0.1:69. Adresu serveru a číslo portu je také možné manuálně zadat. Funkce *getaddrinfo()* umožňuje naplnit strukturu *addrinfo*, která reprezentuje server. Funkce *sendto()* a *recvfrom()* jsou využity pro samotnou komunikaci se serverem - zasílání a přijímání TFTP paketů.

Komunikace začíná zasláním požadavku na čtení nebo zápis souboru. Pokud adresa TFTP serveru není správná, klient zůstane v zablokovaném stavu a uživatel musí program ukončit. V opačném případě server požadavek přijme a pošle odpověď klientovi. Transfer size option je součástí každého požadavku. Další možnosti jsou block size option a timeout option (specifikuje uživatel v zadávacím promptu). V případě akceptování voleb server pošle Option Acknowledgement (OACK) paket klientovi, čímž je spojení navázáno.

```
79 Read Request, File: man.txt, Transfer type: octet, tsize=0, blksize=100, timeout=10
67 Option Acknowledgement, tsize=553, blksize=100, timeout=10
```

Obrázek 1: Figure 1. Zachycené pakety RRQ a OACK ve Wiresharku

Pokud server neakceptuje nějaký option, ten je vynechán z OACK paketu. V případě block size option maximální velikost datového bloku pro přenos je nastaven na 512 B a v případě timeout option, timeout při přenosu se nebude uvažovat.

2.2.1 Čtení

Klient zašle potvrzovací (ACK) paket jako odpověď na OACK, čímž může začít přenos souboru. Server pošle blok dat podle nastavené velikosti, klient musí každý datový paket potvrdit zasláním ACK.

Tok přenosu je řízen pořadovým číslem datového bloku. Pro uložení přijatého paketu existuje buffer, který funkce *fwrite()* vypíše do daného souboru (s korekci +4 pro vynechání TFTP hlavičky). Pokud je velikost dat menší než nastavený blocksize, klient pošle poslední potvrzovací paket a server spojení ukončí.

2.2.2 Zápis

Klient po obdržení OACK začne číst data ze souboru, který chce poslat. To je možné pouze v případě, že daný soubor existuje a je přístupný - jinak následuje chybová hláška. Při zápisu server musí každý přijatý datový paket potvrdit, se správně nastaveným pořadovým číslem.

```
>-w -d obrazek.png -s 1500
Block size is limited to range [8 B - 1468 B].
Maximum transfer block size is set to 1468.
[2021-11-15 22:41:43.805] Requesting WRITE from server 127.0.0.1:69
[2021-11-15 22:41:43.806] Sending DATA ... 1468 B of 5078 B
[2021-11-15 22:41:43.806] Sending DATA ... 2936 B of 5078 B
[2021-11-15 22:41:43.806] Sending DATA ... 4404 B of 5078 B
[2021-11-15 22:41:43.807] Sending DATA ... 5078 B of 5078 B
[2021-11-15 22:41:43.807] Transfer succesfully completed.
```

>|

Obrázek 2: Figure 2. Příklad zápisu souboru na server

3 Testování

Za účelem testování protistrany jsem použil [TFTPD32](#) a [Rust TFTP Server](#). Pro analýzu a sledování toku přenosu TFTP paketů jsem použil Wireshark.

Reference

- [1] The BSD UNIX Socket Library
<https://support.sas.com/documentation/onlinedoc/sasc/doc/lr2/lrv2ch15.htm>
- [2] RFC 1350 *THE TFTP PROTOCOL (REVISION 2)*
- [3] Extending TFTP
<https://www.compuphase.com/tftp.htm>
- [4] RFC 1785 *TFTP Option Negotiation Analysis*
- [5] RFC 2347 *TFTP Option Extension*
- [6] RFC 2348 *TFTP Blocksize Option*
- [7] RFC 2349 *TFTP Timeout Interval and Transfer Size Options*