Vysoké Učení Technické v Brně

Fakulta informačních technologií

Network Applications and Network Administration

2021/2022

Variant: TFTP Client (dr. Veselý)

Nikita Zhukov (xzhuko01)

Brno, November 15, 2021

Contents

1 Introduction

The purpose of project was to implement Trivial File Transfer Protocol (TFTP) client which can communicate with the TFTP server via requesting reading files from the server or writing them into the server.

2 How to run

Program works and has been tested on MacOs 10.15+(MacOS Catalina).**Makefile** has been used in order to build the program.

Steps to run:
1. Build the program:
```
$ make
```
2. Run program:
```
$ ./tftp-client -R/-W -d ABSOLUTEFILEPATH {-t -s -m -c -a}
```
Program supports following arguments:

- **-R** or **-W** Stangs for opcode operation. If *-R* is mentioned then the file from the server will be read. In case of *-W* the file will be rewritten on the server side. One of those arguments should appear. However, only from them can be used.
- **-d** *filename* Defines the the absolute path of the file which is going to be read or written from/to the sever.
- Optional arguments:
  - **-t** (Defines timeout for the server which might be not accepted by its side. By default this option is disabled.)
  - **-s** (Defines maximum size of the package which client offers for sending to the server. Default size of block is set to 512 B.)
  - **-m** (Enable multicasting within transporting. By default this option is disabled.)

  - **-c** *mode* (Defines the mode of transporting data ("*octet*" or "*netascii*"). Default valus is "octet".)
  - **-a** *ip,port* (Defines the address to which TFTP Client is going to connect and send/receive data to/from. Default value is set to IPV4 "localhost" with port 69.)

In case of success computation the program returns **0**. Otherwise might return:

- **1** - Argument's Parsing Error
- **2** - Internal Error

3 Structure of Program

Trival File Transfer Protocol is implemented in C/C++ using following libraries:

- C/C++ libraries
  - **iostream**
  - **cstdio**
  - **string**
  - **getopt.h**
  - **string.h**
  - **netdb.h**
  - **sys/stat.h**
  - **unistd.h**
  - **ctime**
- Internal implementation of the TFTP class
  - **tftp-client.hh**

The files used for implementation:

- **argparser.cpp** - Parser for arguments
- **main.cpp** - Main file
- **tftp-client.cpp** - TFTP Client Class declaration
- **tftp-client.h** - TFTP Client Class definition

4 Implementation

The implementation of TFTP was divided into parts where the first one was resposible for the parsing arguments from CLI and setting some of the filters for getting a desired tftp respond. While the other one was connecting and receiving DATA or ACK packets with all requested information from the server. As well as dealing with error states which might occur during requesting and responding from the server side as well as from the client side. Testing most of the time has been carried out on Mac Os System which is built on part of the unix system.

5 TODO

Due to lack of time during semester the following parts have not been implemented or thoroughly tested.

- **IPV6** - Handling for IPV6 protocol
- **Multicasting**
- **Timeout**
- Parameter **-a**

6 Literature & Inspiration

[1] - Code inspiration `https://www.linkedin.com/pulse/tftp-client-implementation-c-sumit-jha/`
[2] - Documentation to tftp protocol `https://datatracker.ietf.org/doc/html/rfc1350`
[3] - `https://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol`
[4] - Documentation to tftp protocol `http://people.na.infn.it/~garufi/didattica/CorsoAcq/Trasp/Lezione9/tcpip_ill/tftp_tri.htm`