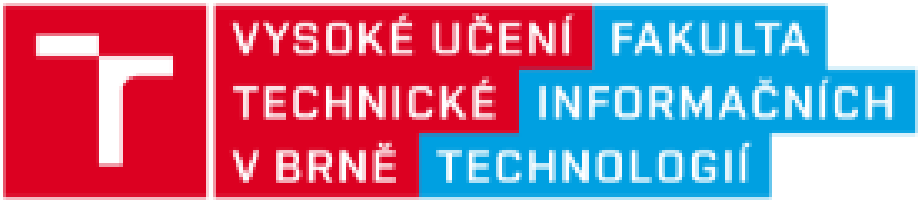

Project documentation for ISA

TFTP Client

Author: Senichak Yahor(xsenic00)

Contents

Introduction

Trivial File Transfer Protocol is a simple protocol that is used to send a file from a server to a client. Trivial File Transfer Protocol uses the UDP concept to exchange messages between server and client. Here UDP stands for User Datagram Protocol.

The goal of the project is to create a TFTP Client with the ability to select the transfer mode (octet or netascii).

1)The default port: 69

2) The default IP : 127.0.0.1

Content of the protocol

The client communicates with the server using packet transmission. One of the tasks is to configure the packages correctly for successful communication. The TFTP protocol contains several types of packets(Standard packet size - 512 bytes):

1) DATA(Data)
2) RRQ(Read Request)
3) WRQ(Write Request)
4) ACK(Acknowledgment)
5) ERROR(Error's)
6) OACK(Option Acknowledgment)

```
Type   Op #     Format without header

           2 bytes     string   1 byte   string     1 byte
-----------------------------------------------
RRQ/ WRQ | 01/02 |  Filename  |   0  |    Mode    |   0  |
    -----------------------------------------------

       2 bytes    2 bytes      n bytes
---------------------------------
DATA  | 03    |   Block #  |    Data    |
---------------------------------

        2 bytes    2 bytes
-------------------
ACK   | 04    |   Block #  |
--------------------


        2 bytes    2 bytes     string    1 byte
---------------------------------------
ERROR | 05    |  ErrorCode |   ErrMsg   |   0  |
---------------------------------------
```

RRQ/WRQ

Bytes:	Function:
[0-1]	Indicate operation(Read [01]- Write[02]- Reserved[00])
[n-bytes]	The name of the file to read or write
[0]	
[n-bytes]	Mod encoding(string "octet" or "netascii")

Data

Bytes:	Function:
[0-1]	Packet ordinal number ("03")
[1-2]	Block number
[n-bytes]	Part of readable or writeable data

ACK

Bytes:	Function:
[0-1]	Packet ordinal number ("04")
[1-2]	Block number

ERROR

Bytes:	Function:
[0-1]	Packet ordinal number ("04")
[1-2]	Code of the error
[n-bytes]	Error message
[0]	

More information(https://datatracker.ietf.org/doc/html/rfc1350)

Implementation

The program consists of main parts: the parser of user arguments(`parser.h`) , the "`Options`" structure which contains all the options specified by the user, packet parser(`paket.h`) , which writes error to STDERR in case of receiving error-paket from server, and the main body of the program containing the main functions (`Read` and `Write`) . Most of the implanted functions and classes related to the control of arguments and packages are called in the body of the main functions(`Read` and `Write`).

In case of the options are missing on the server, the options are reconfigured according to the server.

The program does not support changes in the size of communication packets (-s), also does not support multicast(-m).

In the case of multicast, the client receives the multicast IP address from the server, further communication is impossible.

How to use it

The program can be translated and run both Linux and Windows.

Example of start for Windows: `./-R-d /home/user/readme.txt -a 147.229.181.2,169 -c octet`

Example of start for Linux:

```
./$mytftpclient
> -R -d /home/user/readme.txt -a 147.229.181.2,169 -c octet
```

Mandatory options:

"-d" - Indicates the name of the file to be read or written.

"-W/-R" - indicates a mandatory operation to be performed.

Optional :

"-c" - defines the encoding method in which the file will be written or read.

"-a" - defines the IP address and port of the server with which it will communicate.(By default : 127.0.0.1,69)

"-t" - indicates the frequency with which the server sends us messages