

Projet TFTP

But du projet

Le projet consiste en l'implémentation du protocole TFTP. Ce protocole est une version light du protocole FTP. Le but de ce projet est d'implémenter une version simplifiée de TFTP en utilisant les méthodes vues en cours. Le protocole devra être implémenté en se basant sur la RFC 1350.

Structure du projet

Le projet comportera deux parties principales: le client et le serveur. Chaque partie aura des dépendances communes.

Architecture du projet

utils.py:

```
Boolean ( success ), String ( err_msg ) : send_file( socket , file_descriptor )

Boolean ( success ), String ( err_msg ) : receive_file( socket , file_descriptor , first_data_blk )

(option) : parse_user_input(args)
```

client.py: *(pseudo code)*

```
parse input
send request
open corrects ports, file descriptors
send_file or receive_file
handle error if any
close everything
```

server.py: *(pseudo code)*

```
start listening
loop:
    open correct ports, file descriptors
    send_file or receive_file
    handle error if any
    close files / correct ports
```

packet.py:

```
Enum : OP_CODE
Enum : ERR_CODE

BinaryString : build_packet_rrq(Filename, mode)

BinaryString : build_packet_wrq(Filename, mode)

BinaryString : build_packet_data(Block_num, data)

BinaryString : build_packet_ack(Block_num)

BinaryString : build_packet_err(Err_code)

For RRQ and WRQ
Op_code, File_name, Mode : decode_packet(BinaryString)

For ACK
Op_code, Block_num : decode_packet(BinaryString)

For DATA and ERR
Op_code, Block_num, Msg : decode_packet(BinaryString)
```