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)