COMP 7005
Assignment 2
Design

Andy Tran
A01266629
October 2nd, 2024

Purpose

- This program has a server-side and client-side code. They accept 1 and 3 arguments respectively, from the command line:
  Server:
    - <port>: The port number on which the server will listen for incoming TCP connections.
  Client:
    - <server_ip>: The IP address of the server.
    - <port>: The port number to connect to on the server.
    - <file_path>: The path to the file whose content will be sent to the server.

- The client will send the file content to the server over a TCP socket. The server will count the number of alphabetic characters in the file and return the count to the client.

Data Types

Arguments

Purpose: To hold the unparsed command-line argument information

Field	Type	Description
file_path	string	The path to the file whose content will be sent to the server.
server_ip	string	The IP address of the server to connect to.
port	int	The port number for the TCP connection.

Settings

Purpose: To hold the settings the program needs to run.

Field	Type	Description
buffer_size	int	The buffer size used for sending/receiving data. Default is 4096 bytes.

Context

Purpose: To hold the arguments, settings, and exit information

Field	Type	Description

Functions

Function	Description
count_alphabetuc_chars	Counts the alphabetic characters in the file content.
get_local_ip	Retrieves the local IP address of server
create_server_socket	Creates and binds the server TCP socket to the specified port.
accept_client_connection	Accepts the client's TCP connection.
receive_file_from_client	Receives the file content from client
send_response_to_client	Sends the character count back to the client
handle_client	Processes the client's request.
start_server	Runs the server handles client requests and cleans.
send_file_content	Sends the file content to the server
receive_response	Receives and displays the server response
start_client	Starts the client sends file content and receive server response
validate_port	Validates the port before being used. Ensures within the range.

Pseudocode

count_alphabetic_chars

Parameters

Parameter	Type	Description
file_data	bytes	The file content received from client.

Return

Value	Reason
letter_ocunt	The number of alphabetic characters in the file content

Pseudo Code

```
count_alphabetic_chars(file_data):

    return the count of alphabetic characters in the file_data
```

get_local_ip

Parameters

Parameter	Type	Description

Return

Value	Reason
local_ip	The server local IP address

Pseudo Code

```
get_local_ip():

    create a socket and connect to an external IP to get local IP
address

    return the local IP address
```

create_server_socket

Parameters

Parameter	Type	Description
port	int	The port number the server will listen on.

Return

Value	Reason
server_socket	The active TCP server socket that is bound to the port.

Pseudo Code

```
create_server_socket(port):

    create a TCP socket using socket.socket()

    set the socket options (reuse address)

    bind the server socket to the specified port using bind()

    start listening for incoming connections with listen()

    return the server_socket
```

accept_client_connection

Parameters

Parameter	Type	Description
server_socket	socket.socket	The server socket that listens for incoming client connections.

Return

Value	Reason
client_socket	This is the client side socket that can be used to communicate.

Pseudo Code

```
accept_client_connection(server_socket):

    Accept an incoming client connection using accept()

    Return the client_socket
```

receive_file_from_client

Parameters

Parameter	Type	Description
client_socket	Socket.socket	The socket connected to the client for receiving and sending data.

Return

Value	Reason
file_data	The accumulated file data received from the client

Pseudo Code

```
receive_file_from_client(client_socket):

    initialize file_data as an empty byte string

    while True:

        receive a chunk of data from the client using recv()

        if no more data is received:

            break out of loop

        append the received data to file_data

    return file_data
```

send_response_to_client

Parameters

Parameter	Type	Description
client_socket	Socket.socket	The socket connected to the client for receiving and sending data.
response	string	The response to send to the client

Return

Value	Reason
file_data	The accumulated file data received from the client

Pseudo Code

```
send_response_to_client(client_socket, response):

    send the response to the client using sendall()
```

handle_client

Parameters

Parameter	Type	Description
client_socket	Socket.socket	The socket connected to the client for receiving and sending data.

Return

Value	Reason
none	none

Pseudo Code

```
handle_client(client_socket):

    receive the file content from the client using
receive_file_from_client()

    try to decode the received file content into text

    count the number of alphabetic characters in the file using
count_alphabetic_chars()

    send the character count back to the client using
send_response_to_client()

    close the client socket




.'
```

start_server

Parameters

Parameter	Type	Description
port	int	The port number the server listens on

Return

Value	Reason
none	

Pseudo Code

```
start_server(port):
    create the server socket using create_server_socket(port)

    while True:
        accept a client connection using
accept_client_connection(server_socket)

        handle the client request using handle_client(client_socket)

        if KeyboardInterrupt exception occurs:
            break

    close the server socket using cleanup_server(server_socket)
```

send_file_content

Parameters

Parameter	Type	Description
client_socket	Socket.socket	The active client socket used to send data to the server
file_path	string	The file path string that the client wants to send

Return

Value	Reason
nothing	none.

Pseudo Code

```
send_file_content(client_socket, file_path):

      verify file is txt

    open the file in binary read mode

    while there is data left to send:
        send chunks of the file content to the server via
client_socket

    close the file after the content is sent

    shutdown the client socket for writing using shutdown
```

receive_response

Parameters

Parameter	Type	Description
client_socket	Socket.socket	The active client socket used to send data to the server

Return

Value	Reason
nothing	none.

Pseudo Code

```
receive_response(client_socket):

    receive the server’s response via client_socket

    decode the response

    print the server's response
```

start_client

Parameters

Parameter	Type	Description
file_path	string	The path to the file whose content will be sent to server
server_ip	string	The IP address of the server to connect to
port	int	The port number for the TCP connection

Return

Value	Reason
nothing	none.

Pseudo Code

```
start_client(server_ip, port, file_path):

    create a client socket using socket.socket()

    connect the client_socket to the server_ip and port

    send the file content to the server using
send_file_content(client_socket, file_path)

    receive and print the server's response using
receive_response(client_socket)

    close the client socket
```

validate_port

Parameters

Parameter	Type	Description
port	int	The port number for the TCP connection

Return

Value	Reason
nothing	none.

Pseudo Code

```
validate_port(port):

    check if port is valid in range of 1024 - 65535
     if not print “Not valid port number please enter valid one”
     exit
```