COMP 7005
Assignment 1
Design

Andy Tran
A01266629
Sept 18th, 2024

Purpose

- This program has a server-side code and client-side code which accept 1 and 2 arguments respectively from the command line:

  Server:

  - <socket_path>

  Client:

  - <socket_path>
  - <file_path>
- They will both create a domain socket and connect at <socket_path>. The client will send a request to the server to see if <file_path> exists on the local system.

Data Types

Arguments

Purpose: To hold the unparsed command-line argument information

Field	Type	Description
file_path	string	The path of the file that would like to be checked
socket_path	string	The socket path that the domain socket would like to be at

Settings

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

Field	Type	Description

Context

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

Field	Type	Description

Functions

Function	Description
create_server_socket	Creates and binds the server socket to the socket path
accept_client_connection	Accepts the clients connection to the socket
handle_client_request	Processes the request and sends a response to client
cleanup_server	Closes the socket and removes the socket path
create_client_socket	Creates and binds the client socket to the socket path
send_file_path	Sends the file path to the server
receive_response	Receive and print the server’s response
cleanup_client	Closes the socket and removes the socket path
validate_socket_path	To validate that the argument passed is a valid socket path

Pseudocode

create_server_socket

Parameters

Parameter	Type	Description
socket_path	string	The file system path where the UNIX domain socket will be created.

Return

Value	Reason
server_socket	The active server socket that binded to the socket path.

Pseudo Code

```
create_server_socket(socket_path):
    check if socket_path exists:
        remove the existing socket file to make clean slate

    create a UNIX domain socket using socket.socket()

    bind the socket to socket_path 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
```

handle_client_request

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 file_path from client_socket

      check if file_path is file exists on local system

            if yes, set response to 'File exists.'

            if no, set response to 'File does not exist.'

      send response to client_socket

      if an exception occurs, print the error

      close the client_socket

.'
```

cleanup_server

Parameters

Parameter	Type	Description
server_socket	Socket.socket	The active server socket that needs to be closed
socket_path	string	The path to the UNIX domain socket file that needs to be removed

Return

Value	Reason
none	This function closes the server and cleans up the socket file

Pseudo Code

```
cleanup_Server(server_socket, socket_path)

      close the server_socket

      check if socket_path exists

            remove the socket file
```

start_server

Parameters

Parameter	Type	Description
socket_path	string	The file system path where the UNIX domain socket is created.

Return

Value	Reason
none	This function runs the server, handles requests, and performs cleanup. Continually listening for client requests only closing upon keyboard interuption to close

Pseudo Code

create the server socket using the create_server_socket(socket_path)

    while True

    accept a client connection using accept_client_connection(server_socket)

    handle the client request using handle_client_request(client_socket)

    if a KeyboardInterrupt exception occurs

        system exit

    close the server_socket and remove the socket file using cleanup_server(server_socket, socket_path)

create_client_socket

Parameters

Parameter	Type	Description
socket_path	string	The file system path of the UNIX domain socket to connect to.

Return

Value	Reason
client_socket	The active client socket connected to the server.

Pseudo Code

create a UNIX domain socket using socket.socket()

connect the client_socket to socket_path

return client_socket

send_file_path

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_path (client_socket, file_path)

    send the encoded file_path to the server through client_socket

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

send_file_path (client_socket, file_path)

    receive data from client_socket

    decode the received data to a string

    print server response

cleanup_client

Parameters

Parameter	Type	Description
client_socket	Socket.socket	The active client socket to be cleaned

Return

Value	Reason
nothing	none.

Pseudo Code

```
cleanup_client(client_socket)

        close client socket using close()
```

start_client

Parameters

Parameter	Type	Description
socket_path	string	The file system path of the UNIX domain socket to connect to.
file_path	string	The file path string to send to the server.

Return

Value	Reason
nothing	none.

Pseudo Code

```
start_client (socket_path, file_path)

    create a client socket using create_client_socket(socket_path)

    try
        send the file_path to the server using send_file_path(client_socket, file_path)

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

    catch any socket error

        print the socket error message

    finally
        close and clean up the client socket using cleanup_client(client_socket)
```

validate_socket_path

Parameters

Parameter	Type	Description
socket_path	string	The file system path of the UNIX domain socket to connect to.

Return

Value	Reason
nothing	none.

Pseudo Code

validate_socket_path (socket_path)

    check if socket_path is a string:

        print error message about it needs to be a string

        exit w status 1

    check if socket_path starts w a /

        if it doesnt then print error message about needs to start w /

        exit w status 1