COMP 7005
Assignment 1

Design

Sumit Khanduri
A01296594
September 21, 2024

Purpose

- This program (client) accepts 1 argument from the command line:
    - <filename>
- It sends the <filename> to the server.

Client.py Functions

Function	Description
printContent	Displays the content of the file in the terminal
main	The main function of the program which connects to the server
connect	Connect to the socket file
send	Send data over the connection
recv	Recieve data over the connection

Pseudocode

printContent

Parameters

Parameter	Type	Description
filename	String	The file to read
data	String	The contents of the file received from server

Return

- nothing

Pseudo Code

```
function printContent ( filename , data)
    print "Filename: " + filename
    print "Content:\n " + data
    return
```

main

Parameters

Parameter	Type	Description
filename	string	The program context

Return

- nothing

Pseudo Code

```
function main()

    create a socket s using AF_UNIX, SOCK_STREAM
    set socketpath to "/tmp/socket"

    set allfiles to command-line arguments starting from index 1

    connect the socket s to socketpath

        for each file in allfiles

            send file name through socket s
            receive response from the server (1024 bytes)
            convert response to file_size (integer)

            set received_content to an empty byte array

            while length of received_content is less than file_size
                receive chunk from socket s (1024 bytes)
                if no chunk is received
                    break the loop
                append chunk to received_content

            call printContent(file, decode received_content to
string)

        print "Closing Connection"
        close socket s
```

connect

Parameters

Parameter	Type	Description
socket	Socket	Socket variable
path	String	The file path to connect to

Return

- nothing

Pseudo Code

```
function connect ( socket , path )
      Try:
            Connect socket to path
      Except:
            Print the error
            Exit the program
```

send

Parameters

Parameter	Type	Description
socket	Socket	Socket variable
data	String	The data to send to the server

Return

- nothing

Pseudo Code

```
function send ( socket , data )
      Try:
            Convert the data into bytes
            Send the data using the socket
      Except:
            Print the error
```

```
            Exit the program
```

recv

Parameters

Parameter	Type	Description
socket	Socket	Socket variable

Return

- String

Pseudo Code

```
function recv ( socket )
      Try:
            Receive the data from the socket
            Return the data
      Except:
            Print the error
            Exit the program
```

Server.py Functions

Function	Description
content_retreiver	Retrieves the contents of the file
unbind	Unbind the socketpath
accept	Accepts the connection
main	The main context of the program

cleanup	Unlink the socket path and closes the connection
bind	Bind the socket to the file path
listen	Make the socket listen for the connections
recv	Recv the data sent over the sockets
send	Send data over the socket

Pseudocode

content_retriever

Parameters

Parameter	Type	Description
filepath	String	The file contents to extract

Return

- String

Pseudo Code

```
function printContent ( filepath )
     open file from filepath
     set the contents of the file to content
     return content
```

unbind

Parameters

Parameter	Type	Description
socketpath	String	The path of the socket file to unbind

Return

- nothing

Pseudo Code

```
function unbind ( socketpath )
     Unlink the socketpath
     Raise exception if socketpath exists
```

accept

Parameters

Parameter	Type	Description
socket	Socket	Socket variable

Return

- connection

Pseudo Code

```
function accept ( socket )
     set connection by accepting the connection
     return connection
```

main

Parameters

Parameter	Type	Description

Return

- nothing

Pseudo Code

```
function main ()
create a socket s using AF_UNIX, SOCK_STREAM

bind the socket to socket_path

start listening for connections

while True:
```

```
    accept a connection from client

    while True:
        receive data from client (file name)

        if data received:
            print "File to read: data"

            get file size
            send file size to client

            open the file
            while there is data to read from file:
                send data to client
        else:
            break

    close connection

end
```

cleanup

Parameters

Parameter	Type	Description
connection	socket	The connection established between client and server
Socket path	string	The file path to which the socket is binded

Return

- nothing

Pseudo Code

```
function cleanup( connection, socketpath )
     Close the connection
     Unlink the socket path
```

bind

Parameters

Parameter	Type	Description
socket	Socket	Socket variable
Socket path	string	The file path to which the socket should be binded to

Return

- nothing

Pseudo Code

```
function bind ( socket , socketpath )
      Try:
            Bind socket to the path
      Except:
            Print error
            Exit program
```

listen

Parameters

Parameter	Type	Description
socket	Socket	Socket variable

Return

- nothing

Pseudo Code

```
function listen ( socket )
      Try:
            Make socket listen for connection
      Except:
`           Print error
            Exit the program
```

Recv

Parameters

Parameter	Type	Description
connection	socket	The connection between client and server

Return

- bytes

Pseudo Code

```
function recv ( connection )
     Try:
          Set data to whatever was received from the connection
          Return data
     Except:
          Print the error
          Exit the program
```

send

Parameters

Parameter	Type	Description
connection	Socket	The connection between client and server
data	string	The data to send over the socket

Return

- nothing

Pseudo Code

```
function send ( connection, data )
     Try:
```

```
      Encode the data in to bytes
      Send the data over the connection
Except:
      Print errors
      Exit the program
```