ConnectXpert

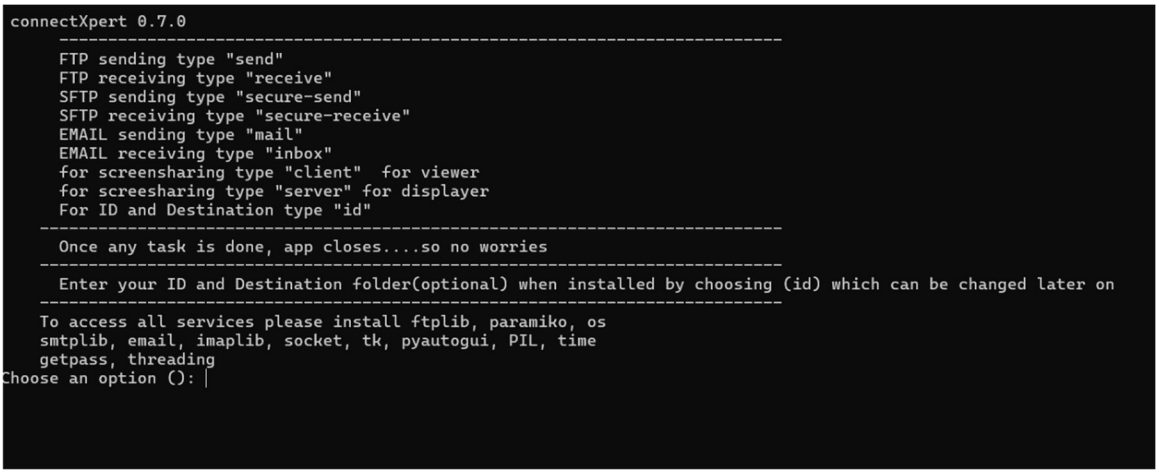
Principles of Computer Systems II

B22AI015
BHALA VIGNESH.CH
B22AI038
VARSHIT MANIKANTHA.S

PROJECT | CSL2090

Libraries Imported

- os, time, socket: Fundamental Python libraries for operating system functionalities, time-related operations, and network communication.
- ftplib, paramiko, smtplib, imaplib: Libraries for FTP, SFTP, SMTP (sending emails), and IMAP (fetching emails).
- tkinter: GUI library for creating a graphical user interface.
- threading: For multithreading support.
- pyautogui, PIL: Libraries for capturing screenshots and handling images.



Functions

We have an extra PC which is just a hardware that can host an OS, so we thought why not make it a SERVER, the ubuntu server on that laptop and its credentials are used here

HOSTNAME, USERNAME, PASSWORD: Credentials for accessing the FTP and SSH of our server.

FTP File Sending and Receiving:

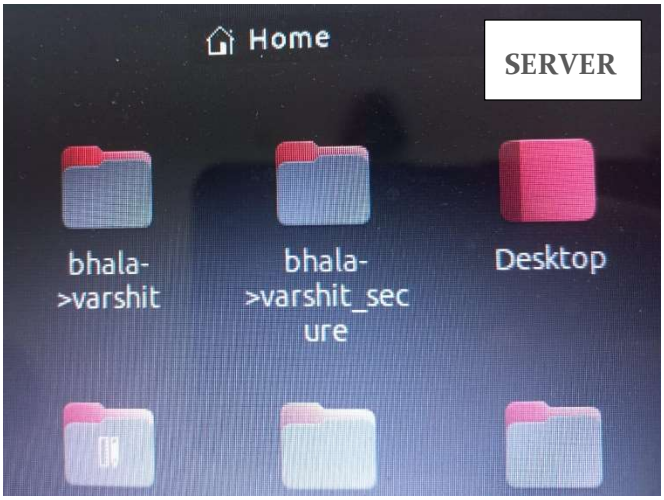
- send_server: Sends a file via FTP to a specified receiver.
- receive_server: Receives files via FTP from a specified sender.

```
Choose an option (): send
Enter receiver name: varshit
Enter file path: "C:\Users\asus\Desktop\connectxpert.ico"
Successfully Sent
```

SFTP File Sending and Receiving:

- sftp_send: Sends a file via SFTP to a specified receiver.
- sftp_receive: Receives files via SFTP from a specified sender.

```
Choose an option (): secure-send
Enter receiver name: varshit_secure
Enter file path: "C:\Users\asus\Desktop\connectxpert.ico"
```



User ID and Directory Registration:

- register_id: Registers the user ID.
- register_dest: Registers the destination directory.

```
Choose an option (): id
setup/change user_id(Y/n): Y
Enter User_ID: bhala
Successfully Regsitered
setup/change directory(Y/n): Y
Enter directory: "C:\Users\asus\Downloads"
Successfully Regsitered
```

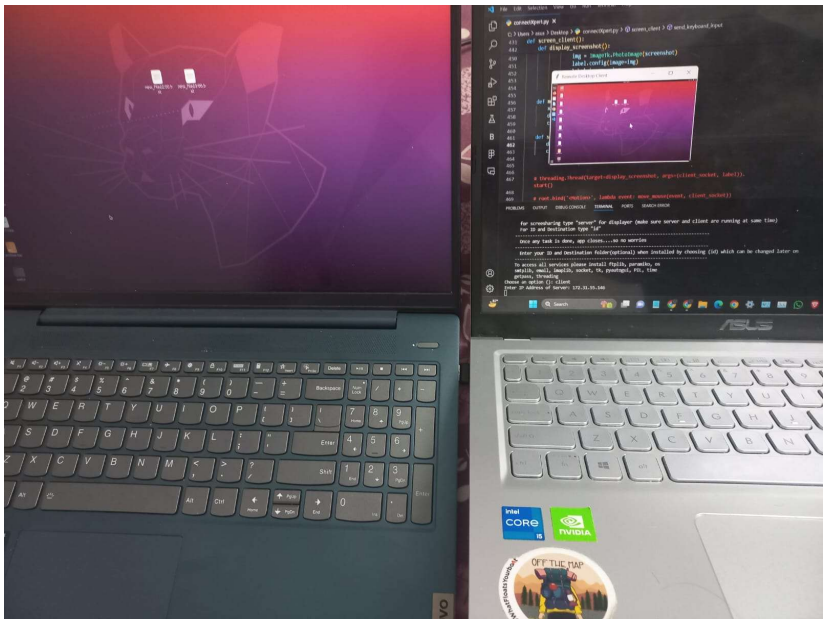
Email Sending and Receiving:

- send_email: Sends an email with optional attachments.
- fetch_emails: Fetches unread emails from a Gmail account.

```
Choose an option (): inbox
Enter your email: b22ai015@iitj.ac.in
Enter your password(APP PASSWORD(Google Authetication)):
From: General Secretary Student Senate <gensecy_ss@iitj.ac.in>
Subject: Re: Nominations for PG and Differently Abled Representatives for
 Student Activity Council: 2024-25
Body: [<email.message.Message object at 0x000001EAC6C884A0>, <email.message.Message object at 0x000001EAC59ADDF0>]
From: Codeforces@codeforces.com
Subject: 2023 Post World Finals Online ICPC Challenge powered by Huawei
Body: [<email.message.Message object at 0x000001EAC6C883E0>, <email.message.Message object at 0x000001EAC6C88530>]
From: Reddit <noreply@redditmail.com>
Subject: "I made Jellyfin resilient - a demo of a thre..."
```

Remote Desktop Sharing:

- screen_share: Sends screenshots of the server's desktop.
- receive_mouse_movement: Receives and processes mouse movements.
- receive_keyboard_input: Receives and processes keyboard input.
- start_server: Starts the server for remote desktop sharing.
- display_screenshot: Displays the received screenshots on the client side.
- move_mouse: Sends mouse movement coordinates.
- send_keyboard_input: Sends keyboard input.



User Interface Functions:

- sftp__receive, sftp__send: Initiates secure file transfer.

- user_id: Manages user ID registration.
- send_mail: Initiates email sending.
- mail_view: Initiates email fetching.
- screen_server: Starts the server for remote desktop sharing.
- screen_client: Initiates the client for remote desktop sharing.

SOCKET

- The socket module allows Python programs to establish connections, send and receive data, and handle network communication.
- It abstracts low-level network operations, making it easier to work with network protocols.
- Sockets can be used for various types of communication, including TCP/IP, UDP, and Unix domain sockets.
- They provide a flexible and powerful mechanism for building networked applications, such as servers, clients, and peer-to-peer systems.

Socket Usage in the Code:

- Server Socket Creation: socket.socket()
- Binding and Listening: bind(), listen()
- Accepting Connections: accept()
- Client Socket Creation: socket.socket()
- Connecting to Server: connect()
- Data Transmission: send(), recv()

- **Remote Desktop Sharing:**
  - The socket module is used to implement remote desktop sharing functionality between a server and a client.
  - In the start_server() function, a server socket is created using socket.socket().
  - The server socket is then bound to a specific IP address and port using bind().
  - It listens for incoming connections using listen().
  - When a client connects, a new socket is created for communication with that client using accept().
  - Communication between the server and client for screen sharing and input is facilitated through these sockets.
- **Screen Sharing Server:**
  - In the start_server() function, the server socket listens for incoming connections.

    - Once a connection is established, a new thread is spawned to handle screen sharing, mouse movement, and keyboard input.
    - The screen_share() function continuously captures screenshots and sends them to the client over the socket.
    - Separate threads handle receiving mouse movement and keyboard input from the client and simulating those actions on the server side.
- **Screen Sharing Client:**
    - In the screen_client() function, the client socket is created to connect to the server's IP address and port.
    - The client receives screenshots from the server and displays them using a Tkinter label.
    - Mouse movement and keyboard input events are captured using event handlers, and corresponding data is sent to the server over the socket.

FTP(File Transfer Protocol)

FTP is a standard network protocol used for transferring files between a client and a server on a computer network. Here's some key information about FTP:

- **Purpose:** FTP is primarily used for transferring files from one host to another over a TCP-based network such as the Internet.
- **Protocol:** It operates on two separate channels: the command channel (for sending commands) and the data channel (for transferring files).
    - Uses TCP ports 20 (data) and 21 (control/command) by default.
- **Security:** Traditional FTP operates in plaintext, which means data, including usernames, passwords, and file contents, are transmitted unencrypted.
    - FTPS (FTP Secure) and FTPES (FTP over Explicit TLS/SSL) are extensions of FTP that add support for encryption using SSL/TLS protocols.

- **Authentication:** Users typically authenticate with a username and password, although anonymous FTP access is also supported.

Code:

- **send_server(file_path, name):**
    - This function is responsible for sending a file via FTP to a specified receiver.
    - Parameters:
    - file_path: The path of the file to be sent.

- name: The name of the receiver.
- It establishes a connection to the FTP server using the provided credentials (HOSTNAME, USERNAME, PASSWORD).
- Constructs the directory structure based on the sender's and receiver's names.
- Uploads the file to the specified directory on the FTP server.
- Handles exceptions such as authentication errors, file not found errors, etc.

- **receive_server(name):**
  - This function is responsible for receiving files via FTP from a specified sender.
  - Parameters:
  - name: The name of the sender.
  - It connects to the FTP server using the provided credentials.
  - Reads the name of the sender from a file (user_id.txt).
  - Reads the name of the receiver from another file (user_dest.txt).
  - Downloads files from the sender's directory on the FTP server to the local machine.
  - Deletes the files from the server after downloading.
  - Handles various exceptions like FTP errors, file not found errors, etc

SFTP(SSH File Transfer Protocol)

SFTP is a secure file transfer protocol that provides file access, file transfer, and file management functionalities over a secure data stream. Here are some key points about SFTP:

- **Security:** SFTP encrypts both commands and data using the Secure Shell (SSH) protocol, providing a secure channel for file transfer and manipulation.
- **Authentication:** SFTP typically uses SSH keys for authentication, providing a more secure method compared to username/password authentication used in traditional FTP.
- **Port:** SFTP usually operates over SSH port 22 by default, although this can be configured.
- **Functionality:** SFTP supports a range of operations, including file uploads, downloads, directory listings, file renaming, and permission changes.

Code:

- **sftp_send(file_path, name):**
  - This function sends a file via SFTP to a specified receiver.
  - Parameters:
  - file_path: The path of the file to be sent.

    - name: The name of the receiver.
    - It establishes an SSH connection to the server using paramiko library.
    - Constructs the directory structure based on sender's and receiver's names.
    - Uploads the file to the specified directory on the server using SFTP.
    - Handles exceptions like authentication errors, file not found errors, etc.
- **sftp_receive(name):**
    - This function receives files via SFTP from a specified sender.
    - Parameters:
    - name: The name of the sender.
    - It establishes an SSH connection to the server using paramiko library.
    - Reads the name of the sender from a file (user_id.txt).
    - Reads the name of the receiver from another file (user_dest.txt).
    - Downloads files from the sender's directory on the server to the local machine using SFTP.
    - Handles various exceptions like authentication errors, file not found errors, etc.

SMTP and IMAP

SMTP and IMAP are protocols used for email communication. Here's an overview of each:

- **SMTP (Simple Mail Transfer Protocol):**
    - SMTP is a protocol used to send email messages between email servers.
    - It operates on TCP port 25 by default.
    - SMTP is a push protocol, meaning it initiates the transfer of email messages from the sender's mail server to the recipient's mail server.
    - SMTP servers are responsible for relaying and delivering email messages to their intended recipients.
- **IMAP (Internet Message Access Protocol):**
    - IMAP is a protocol used by email clients to retrieve email messages from a mail server. It operates on TCP port 143 by default (IMAP4), or on port 993 for IMAP over SSL/TLS (IMAPS).
    - IMAP allows users to access their email messages stored on a remote mail server and manipulate them without downloading
    - IMAP servers maintain the state of the user's email folders, including read/unread status, flags, and folder structure.