Documentations

Some information before you start (what I've done):

- a client code (can connect to several servers)
- a server code (accepts several clients, to change the ip and the port you have to do it directly in the code)

Server's documentation

User Documentation

This server application allows clients to connect and send commands to be executed on the server. The server can handle multiple client connections simultaneously. The available commands are:

- exit: Close the connection and shut down the server.
- disconnect: Close the connection with the current client, but keep the server running.
- kill: Shut down the server.
- reset: Restart the server.
- os: Send the operating system of the server to the client.
- ip: Send the IP address of the server to the client.
- hostname: Send the hostname of the server to the client.
- ram: Send information about the usage and availability of RAM on the server to the client.
- cpu: Send information about the CPU of the server to the client.

To use the server, run the server.py file. The server will start listening for client connections on port 10000.

Technical Documentation

The server code uses the following libraries:

- socket: For creating and managing network sockets.
- threading: For creating and managing threads.
- platform: For determining the operating system of the server.
- subprocess: For running command-line processes.
- os: For interacting with the operating system.
- time: For pausing the execution of the code.
- psutil: For getting information about the server's hardware and resources.
- cpuinfo: For getting information about the server's CPU.

The server has two main functions: accept_clients and handle_client.

The accept_clients function creates a socket and listens for client connections on port 10000. When a client connects, a new thread is created to handle the client using the handle_client function.

The handle_client function receives data from the client and executes the appropriate action based on the command received. If the command

Installation Guide

1. Make sure you have Python 3 installed on your machine. You can check if you have Python installed by running the following command in a terminal:

```
python3 --version
```

2. If you don't have Python installed, you can download and install it from the official website: https://www.python.org/downloads/
3. Clone the repository or download the server.py file.
4. Install the required libraries. You can do this by running the following command in a terminal:

```
pip install -r requirements.txt
```

5. Run the server by executing the server.py file:

```
python server.py
```

The server will start listening for client connections on port 10000. You can now connect to the server using a client application.

Client’s documentation

User Documentation

The client application is a GUI that allows you to connect to a server and send commands to it.

To use the client, follow these steps:

1. Enter the IP address and port of the server you want to connect to in the "Adresse IP" and "Port" fields, respectively.

2. Click on the "Connexion" button to establish a connection to the server. If the connection is successful, the "Status" field will display "Connecté" and a new window will open.
3. In the new window, you can enter commands in the input field at the bottom and press the "Envoyer" button to send them to the server. The server's response will be displayed in the text area above.
4. To disconnect from the server, click on the "Déconnexion" button in the main window. The "Status" field will display "Déconnecté" and the connection will be terminated.

Technical Documentation

The client code is written in Python 3 and uses the PyQt5 library to create a GUI.

The MainWindow class represents the main window of the application, and contains the following elements:

- A QLineEdit widget for the IP address of the server.
- A QLineEdit widget for the port of the server.
- A QPushButton widget for connecting to the server.
- A QLabel widget for displaying the connection status.
- A QPushButton widget for disconnecting from the server.

The Shell class represents the window for sending commands to the server and receiving responses. It contains the following elements:

- A QTextEdit widget for displaying the server's responses.
- A QLineEdit widget for entering commands to be

Installation Guide

1. Make sure you have Python 3 and PyQt5 installed on your machine. You can check if you have Python and PyQt5 installed by running the following commands in a terminal:

python3 --version

pip3 show PyQt5

2. If you don't have Python installed, you can download and install it from the official website: https://www.python.org/downloads/. If you don't have PyQt5 installed, you can install it using the following command:

pip3 install PyQt5

3. Clone the repository or download the client.py file.
4. Install the required libraries. You can do this by running the following command in a terminal:

```
pip install -r requirements.txt
```

5. Run the client by executing the client.py file:

```
python client.py
```

The client application will open, and you can enter the IP address and port of the server you want to connect to, and click on the "Connexion" button to establish a connection. You can then send commands to the server and receive responses.