

Servidor e Cliente Python com Comunicação via Sockets

Este é um projeto de exemplo de um sistema cliente e servidor Python que se comunicam usando sockets TCP/IP. O projeto consiste em dois scripts: um para o servidor e outro para o cliente.

Servidor

O código do servidor está no arquivo `main.py`. Ele utiliza a biblioteca `socket`, `threading` e outras para criar um servidor que pode atender múltiplos clientes simultaneamente. O servidor oferece os seguintes serviços:

- Listar arquivos em um diretório.
- Enviar imagens para clientes.
- Enviar poesias de texto para clientes.
- Fornecer a hora do servidor.
- Encerrar a conexão com os clientes.

Para iniciar o servidor, siga estas etapas:

- 1.Abra o arquivo `main.py` em um ambiente Python.
- 2.Certifique-se de que a porta e o endereço IP do servidor estejam configurados corretamente na linha:

```
server_socket.bind(('127.0.0.1', 12345))
```

Você pode modificar o endereço IP e a porta conforme necessário.

- 3.Execute o servidor executando o arquivo `main.py`.

Cliente

O código do cliente está no arquivo `client.py`. Ele utiliza a biblioteca `tkinter` para criar uma interface gráfica simples. O cliente permite que o usuário:

- Obtenha a hora do servidor.
- Receba imagens do servidor.
- Receba poesias de texto do servidor.

- Liste arquivos no servidor.
- Encerre a conexão com o servidor.

Para configurar e executar o cliente, siga estas etapas:

- 1.Abra o arquivo `client.py` em um ambiente Python.
- 2.Verifique se o cliente está configurado para se conectar ao mesmo endereço IP e porta que o servidor está ouvindo na linha:

```
cliente_socket.connect(('localhost', 12345))
```

Certifique-se de que os valores correspondam aos configurados no servidor.

- 3.Execute o cliente executando o arquivo `client.py`.

Uso do Cliente

Após iniciar o cliente, você verá um menu simples. Você pode inserir o nome de um arquivo (por exemplo, "poema") para solicitar um arquivo de texto do servidor. O servidor procurará o arquivo no diretório "arquivos" e o enviará de volta ao cliente se existir.

Você também pode digitar "exit" para encerrar a conexão com o servidor e sair do cliente.

Notas Adicionais

- Os arquivos de texto solicitados pelo cliente são armazenados na pasta "arquivos" do servidor e enviados ao cliente quando solicitados.
- O servidor é capaz de lidar com várias conexões de clientes simultaneamente por meio de threads.
- Certifique-se de que os arquivos de texto existam no diretório "arquivos" antes de solicitar seu envio ao servidor.

Este é apenas um exemplo simples de comunicação cliente-servidor usando sockets e pode ser usado como base para construir aplicativos mais complexos.

Desafios

- Tivemos problema no envio de arquivos jpg para o cliente pois ele enviava o arquivo particionado e o cliente só recebia somente um pedaço da imagem.
- Implementamos que o cliente receberia os pedaços, os uniria e carregava a imagem.

Equipe

Billy Grahlan Alves Rodrigues - 508010

João Victor dos Santos Ferreira - 519334