

## Simple File Server (Server)

### Problem Statement:

You are tasked with writing a simple file server and a client program that can download files from the server.

### Server Requirements:

The server should:

- Listen on localhost (IP address 127.0.0.1) and port 12345.
- Accept a single connection at a time.
- When a client connects, read the requested filename from the client.
- Look up the file in an internal dictionary of files and their contents.
- Send the content of the requested file back to the client.
- If the requested file is not found, send the message "File not found." back to the client.
- Close the connection after sending the file content or the "File not found." message.

### Client Requirements:

The client program should:

- Take the filename as a command-line argument.
- Connect to the server on localhost and port 12345.
- Send the filename to the server.
- Receive the file content (or the "File not found." message) from the server.
- Print the received content to the console.
- Close the connection to the server.

### Input:

Your server program should not require any input.

Your client program should take the filename as a command-line argument.

### Output of the server:

```
Testing file download existing ...
Connected by ('127.0.0.1', 12345)
recv called with: call(1024)
sendall called with: call(b'Hello, this is the content of
example.txt')
close called with: call()
Testing file download not exist ...
Connected by ('127.0.0.1', 12345)
recv called with: call(1024)
sendall called with: call(b'File not found.')
close called with: call()
```

### Output of the client:

```
Testing file download not exist ...
Connected to server on port 12345
Received from server:
```

File not found.  
Connection closed.  
connect called with: call(('127.0.0.1', 12345))  
sendall called with: call(b'non\_existent\_file.txt')  
recv called with: call(1024)  
Testing file download success ...  
Connected to server on port 12345  
Received from server:  
Hello, this is the content of example.txt  
Connection closed.  
connect called with: call(('127.0.0.1', 12345))  
sendall called with: call(b'example.txt')