RPC File Transfer System

Le Anh Quang

Design of the RPC Service

The RPC service is implemented using XML-RPC, a remote procedure call protocol that uses XML to encode calls and HTTP as a transport mechanism. The server provides three main functionalities:

- upload(fileName, content): Upload files to the server.
- download (fileName): Download files from the server.
- list(): List all files available on the server.

The system is designed with a server-client model:

- The server hosts the files and handles client requests.
- The client interacts with the server to perform file operations.

System Organization

The server and client scripts are written in Python.

- The server script (`server.py`) sets up the RPC server and registers functions for file operations.
- The client script (`client.py`) interacts with the server via an XML-RPC proxy.

Implementation of the File Transfer

The following code snippets showcase the implementation of key functionalities:

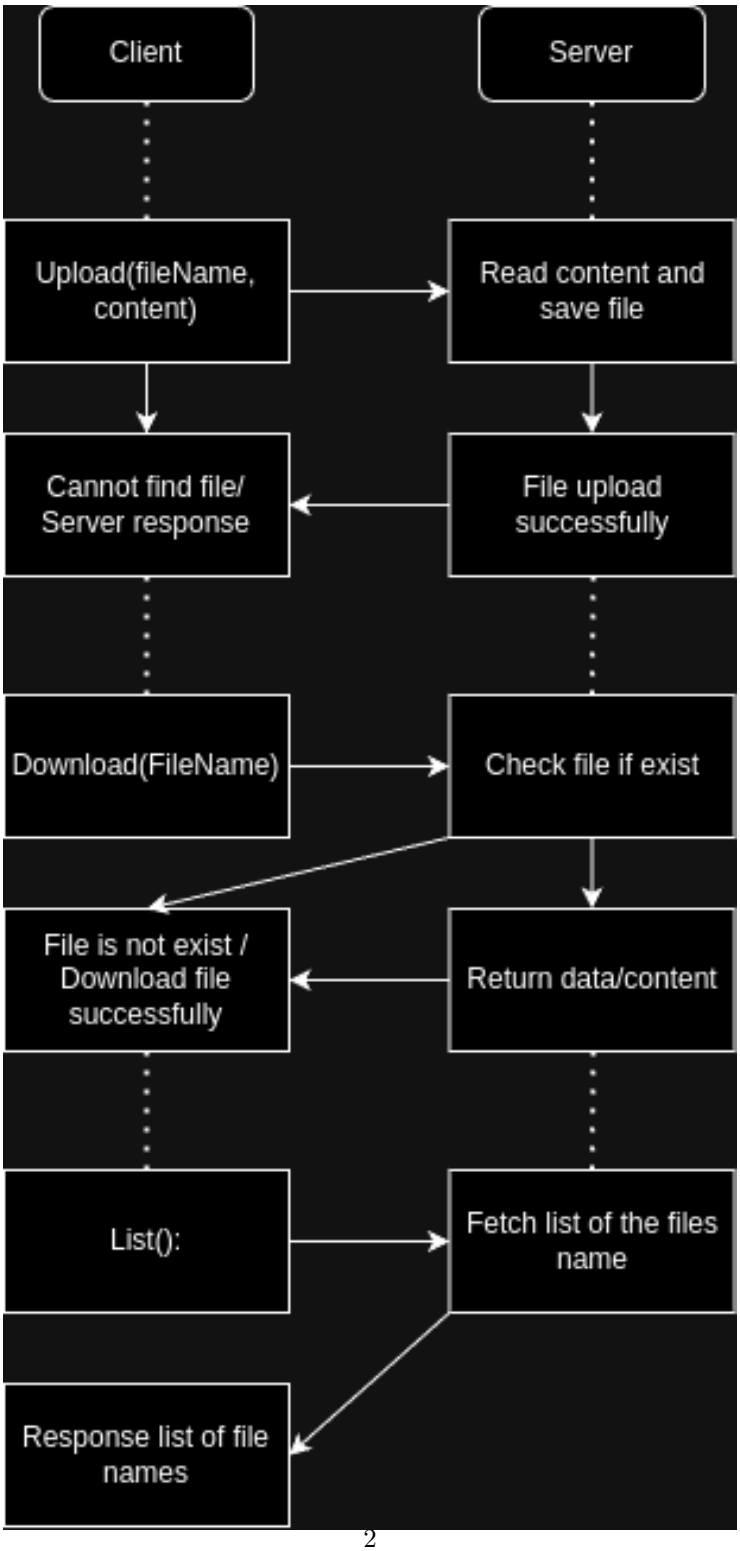

Figure 1: High-Level Design of the RPC Service

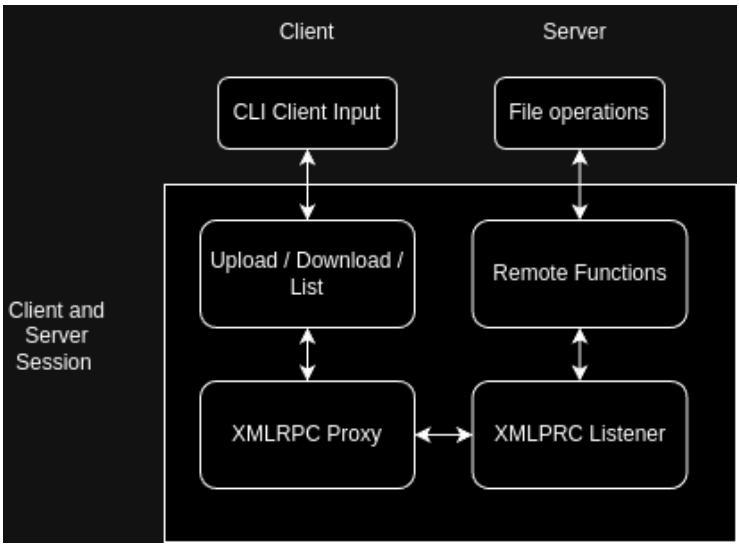

Figure 2: System Organization Diagram

Client side:

```
import xmlrpc.client
import os
from typing import Dict, Callable


def upload(argument):
        server = argument[0]
        filename = argument[1]

        if not os.path.exists(filename):
                return "File does not exist"

        with open(filename, "rb") as file:
                content = file.read()

        response = server.upload(filename, content)

        return response


def download(argument):
        server = argument[0]
        filename = argument[1]
        content = server.download(filename)

```

```
        if not content:
                return "File not found"
        else:
                with open(filename, "wb") as f:
                        f.write(content.data)
        return f"File {filename} downloaded successfully."


def list(argument):
        server = argument[0]
        files = server.list()
        return files


if __name__ == "__main__":
        host = ""
        port = 8080

        server = xmlrpc.client.ServerProxy(f"http://{host}:{
            port}/", allow_none=True)
        options = {
                "UPLOAD": upload,
                "DOWNLOAD": download,
                "LIST": list,
        }
        while True:
                try:
                        userInput = (
                        input("Enter operation (UPLOAD <
                            file_name>, DOWNLOAD <file_name>,
                             LIST) or QUIT to exit: ").strip
                            ()
                        )
                        userInput = userInput.split(" ")
                        while userInput.count(" "):
                                userInput.remove(" ")
                        operation = userInput[0].upper()
                        argument = userInput[-1 : ]

                        argument = [server] + argument

                        if operation == "QUIT":
                                break

                        if operation not in options:
                                print("Invalid operation")
                                continue
                        else:
                                response = options[operation
                                    ](argument)
```

```
                                print("Response from server:
                                    ", response)

                except Exception as e:
                print(f"Error: {e}")
                continue
```

Listing 1: Client-side Upload Function

Server Side:

```
import xmlrpc.server
import os

def download(fileName):
        if (os.path.exists(fileName)):
                with open(fileName, 'rb') as f:
                        content = f.read()
                return content
        else:
                return False

def upload(fileName : str, content : bytes):
        with open(fileName, "wb") as f:
                f.write(content.data)
        return "Successfully"

def list():
        fileList = os.listdir()
        return fileList

def main():
        host = "192.168.127.103"
        port = 8080
        server = xmlrpc.server.SimpleXMLRPCServer((host,
            port), allow_none=True)
        print("Start the server")
        server.register_function(upload, "upload")
        server.register_function(list, "list")
        server.register_function(download, "download")
        server.serve_forever()

        return

if __name__ == "__main__":
        main()
```

Listing 2: Server Main Function