

Application Protocol

Overview

This document outlines the application protocol used in a client-server program for performing simple checks and calculations with Integers. The protocol is designed for communication over TCP between the server and multiple clients.

Design Overview

Server

The server is implemented in Python and follows a concurrent design utilizing the select module to efficiently handle multiple client connections simultaneously, and supports various commands.

The server processes commands such as calculations, palindrome and primality checks. To use the server, the client must authenticate first.

Client

The client is implemented in Python and provides a command-line interface for users to interact with the server.

The command-line interface supports all the services provided by the server and authenticates clients as well.

Application Protocol Definition

1. Message Format:

Data exchanged between the client and server is sent and received in two different messages and follow the format:

- <message_length>#<message_body>
 - <message_length> is a 4-byte unsigned integer representing the length of the <message_body>.
 - <message_body> contains the actual content of the message.

* The message is sent and received as one string

2. Message Types:

Client-to-Server Messages:

Username and Password Requests:

- `<message_body>`: The username / password entered by the client.

Service Request:

- `<message_body>`: The service requested by the client (e.g., "calculate: 5 + 3").

Server-to-Client Messages:

Welcome Message:

- `<message_body>`: A welcome message sent by the server when the client initially connects.

Username Format Response:

- `<message_body>`: Status message indicating whether the username format is valid.

Login Response:

- `<message_body>`: Status message indicating whether the login was successful or failed. If login was successful, the body contains a greeting message to the client.

Service Response:

- `<message_body>`: The result of the requested service or an error message.

3. Data Encoding:

All data within `<message_length>#<message_body>` is binary encoded.

4. Connection Management:

The connection is established using TCP/IP.

5. Command Set:

Client-to-Server Commands:

User: <username>: Sends the username for authentication.

Password: <password>: Sends the password for authentication.

<service>: <service_params>: Requests the service <service> with the provided parameters <service_params>.

Client-Only Commands

quit: Request to quit from the session.

Documentation

Connection and Login

Upon successful connection to the server, the client is greeted with a welcome message. The client must then log in by providing a valid username and password in the following format:

User: <username>

Password: <password>

- <username>: The username of the client.
- <password>: The password of the client.

Response:

If login is successful:

Hi <username>, good to see you.

If login fails:

Failed to login.

Supported Commands

1. Calculate Command:

calculate: <x> <y> <z>

- <x>, <z>: Integers.
- <y>: One of the following operations: "/", "x", "-", "+"

Response:

response: <r>

- <r>: The result of the calculation.

2. Palindrome Check Command:

is_palindrome: <x>

- <x>: The number to check for palindrome.

Response:

response: Yes / No

- Yes: If <x> is a palindrome.
- No: If <x> is not a palindrome.

3. Primality Check Command:

is_primary: <x>

- <x>: The number to check for primality.

Response:

response: Yes / No

- Yes: If <x> is a prime number.
- No: If <x> is not a prime number.

Closing the Session

To terminate the session, the client can enter the command:
quit

Error Handling

If the server receives unexpected input or encounters an error, it will print an error message and disconnect the client.

Running the Programs

To run the server:

`./numbers_server <users_file> <port>`

- `<users_file>`: The file that contains usernames and their corresponding passwords. Each line contains a username and its password, tab delimited.
- `<port>`: The port to listen to. It's optional. Default value: 1337

To run the client:

`./numbers_client <hostname <port>>`

The `<hostname>` and `<port>` are optional. `<hostname>` cannot be provided without `<port>`

- `<hostname>`: The hostname or IP address of the server to connect to.
Default value: 'localhost'
- `<port>`: The port number on which the server is listening for connections.
Default value: 1337