VUT FIT 2017/18

IPK Project 1 Documentation

Client-server application for retrieving user data

by

Samuel Bohovic
xbohov01@stud.fit.vutbr.cz

1.3.2018 Brno

Table of Contents

1. Introduction

The goal of this project was to develop an application that would allow user to retrieve information about users and user folders from a remote server. Application consists of two programs. The client application that provides user with an interface to contact the remote server. The second program is the server application which receives demands from client application, reads information about users and sends this information back to the client.

2. Theory and requirements

At first relevant information about programming of network applications had to be acquired. Most of this information was provided in LINUX manual pages, lecture presentations and online sources.

Secondly, because creating a proprietary communication protocol was required, some time was dedicated to studying existing protocols, such as TCP, UDP and SMTP.

3. Communication protocol

3.1. Client side

Client sends one message and receives two.

First message is composed of a random number, this numbers hash and data request. Data request is contains the kind of data requested (name, home directory, list) and a login. Hash in the message serves for authorization by the server.

First message received contains a single number. The length of data to be sent by the server. This is to that client can allocate enough space for the data.

Last received message contains the data itself. Data is printed on standard output. After this client's socket is closed.

3.2 Server side

Server runs in an endless loop waiting for clients to connect. When client connects a child process is forked to handle client's request.

Upon connection server receives initial message from client. First it reads number and hash from this message. Number is hashed and compared to received hash. If they are not matching server ends the connection and child process exits. Otherwise data request part of message is parsed.

Server loads data from /etc/passwd file and reads the length of data. Lenght of data is sent to client. Server then waits for one second to allow message to be processed by the client.

Finally data is sent to client, server again waits one second and then closes the connection. Memory is freed and child process exits.

Parent process doesn't monitor child processes in any way. It only serves to create them.

4. Known issues

- When -l flag is used, not all data may be received (works reliably up to 8000 characters).

5. Literature

1) Understanding /etc/passwd, Online: https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/
2) Kurose J.F., Ross K.W.: Computer Networking, A Top-Down Approach Featuring the Internet. Addison-Wesley, 2012.
3) Designing and Implementing an Application Layer Network Protocol Using UNIX Sockets and TCP, Online: http://www.egr.msu.edu/classes/ece480/capstone/fall12/group02/documents/Ryan-Lattrel_App-Note.pdf
4) On the Design of Application Protocols, Online: https://tools.ietf.org/html/rfc3117