Documentation of the application layer of the ISA project 2016/2017
Name and surname: Jakub Vitásek
Login: xvitas02

1 Assignment

The assignment was to create an IMAP4rev1 (RFC 3501) client using TLS by utilizing the OpenSSL library. The program downloads messages saved on the specified server, provided it is supplied with the appropriate and valid arguments, and saves the messages to a specified directory. The program also outputs the number of downloaded messages.

2 Implementation

The program uses a separate module (class `input_parser`), which calls an internal private method upon being constructed. The method parses the arguments and saves them to class attributes while maintaining default values when unspecified.

Once the arguments are parsed the program checks if TLS should be used. This branches the application into two parts, where the appropriate module is called – either class `imap` or class `imaps`. Either way, the logical and the modular part of the implementation stays exactly the same.

2.1 Sending commands to the server

The `imap/imaps` module utilizes multiple IMAP commands to get to the data requested by the user. It uses LOGIN after establishing a TCP/IP connection with the server. TCP is more appropriate for this application, owing to its maintaining the connection open until closure. After being successfully authenticated, the program sends a SELECT command to enter the mailbox specified by user parameters.

2.2 Fetching the messages

Afterwards, the IMAP module sends multiple commands to fetch UIDs of all messages, either only the unseen or all of them. The response is parsed into a vector of integers which is iterated over by the `fetchAllMessages` method. When a message is parsed into a final string, the string is passed to a `saveMessage` method which handles the path to the output directory and the actual saving of the file.

2.3 Cleaning up and error handling

When the whole process is completed, the program outputs the number of downloaded messages. The output is modified if the option to download only unseen messages is requested. The application returns 0, if everything went without a problem.

There is a separate module to handle errors. Each error has a unique code with which it exits, while printing a specified message to inform the user.