

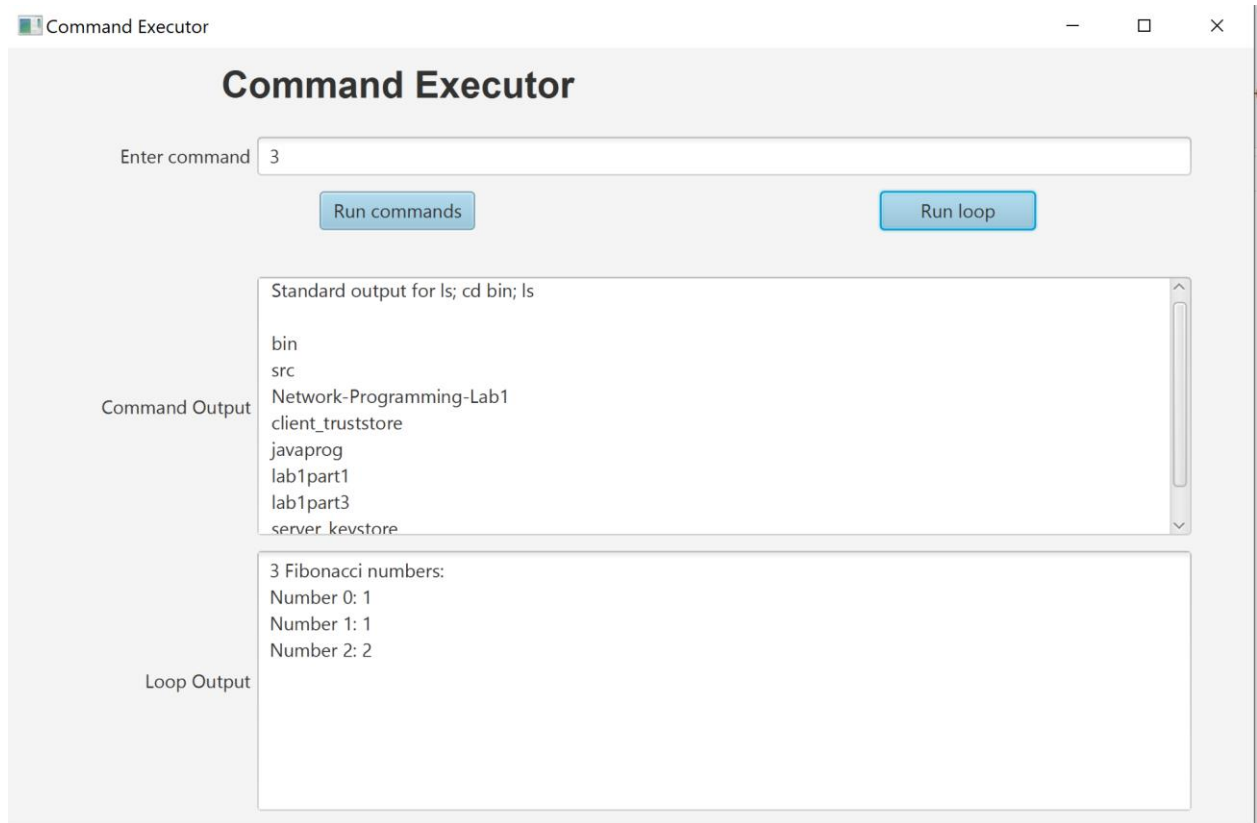
Lab 2 report

Part 1

GUI enables entering command or number of Fibonacci values and two separate buttons.

Every command starts a new thread, so, while numbers are being computed, it is possible to execute OS commands with “Run commands” button.

Command line is started automatically depending on OS type (Windows/Linux).



Part 2

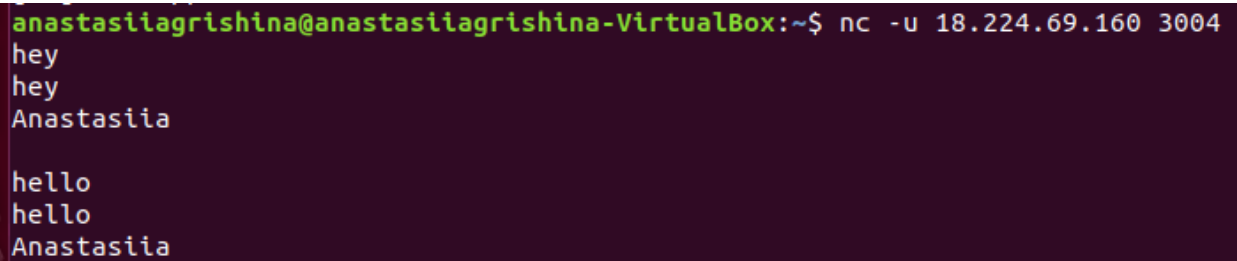
Lookup – returns IP addresses of consecutively passed hostnames in the arguments

```
ubuntu@ip-172-31-3-115:~/jar-files$ java -jar Lookup_lab1part2.jar google.com
google.com:172.217.0.14. Name: Anastasiia
ubuntu@ip-172-31-3-115:~/jar-files$ java -jar Lookup_lab1part2.jar google.com facebook.com
google.com:172.217.0.14. Name: Anastasiia
facebook.com:31.13.71.36. Name: Anastasiia
ubuntu@ip-172-31-3-115:~/jar-files$
```

TCPEchoServer – establishes TCP connection, gets the client input and sends it back. For appending a name, it should be converted to bytes and concatenated with the byte array of client input. Tested on AWS instance

```
C:\Users\agari>telnet 18.224.69.160 9000
Trying 18.224.69.160...
Connected to 18.224.69.160.
Escape character is '^]'.
hey
Anastasiia hey
How are you
Anastasiia How are you
```

UDPEchoServer – establishes UDP connection, gets the client input and sends it back. For appending a name, it should be converted to bytes and concatenated with the byte array of client input. Datagrams should maintain accurate length and client address, if a new datagram is created. Tested on AWS and Ubuntu VM



```
anastasiagrishina@anastasiagrishina-VirtualBox:~$ nc -u 18.224.69.160 3004
hey
hey
Anastasia

hello
hello
Anastasia
```

Race0

- a) Unpredictable strings of X and dots, since the shared resource is asynchronously accessed by two threads
- c) Dots are displayed, since both sleep and increments are implemented and accessed synchronously.

Part 3

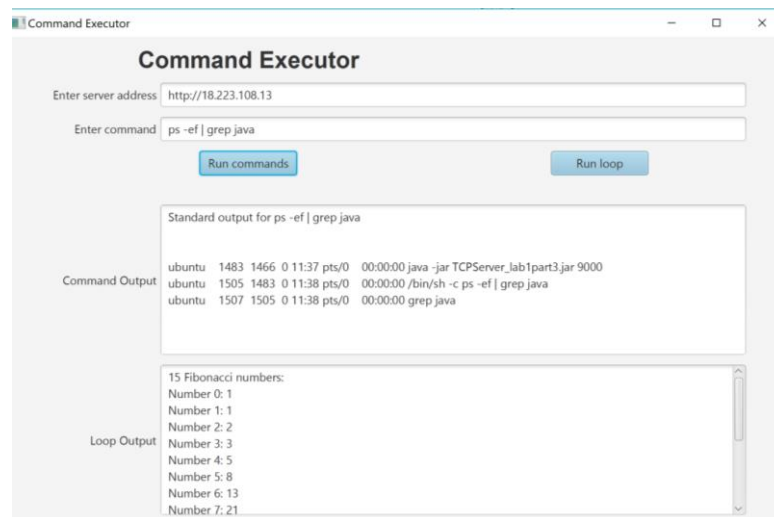
Version 1.

Two buttons initialize parallel client threads, where a URL, a command type (loop or OS command), command text, and text area pointer are passed. Clients connect to the multithreaded server. On the server a listener is implemented. With every new connection a new thread is started on server, server Runnable implements run method, which executes commands and sends response – standard output or standards errors, or Fibonacci numbers. Client reads the response and updates text in the text area

Version 2.

Two buttons send http GET requests passing a URL, a command type (loop or OS command), command text to a method of HttpRequest class. In HttpRequest class, the request is formed from the passed parameters and sent to the server. A multithreaded server starts new thread for every TCP connection and runs a CmdWrkrRunnable class run method. The method decodes url, extracts the query, executes corresponding commands and forms an encoded response with the output of the commands execution. Server response is read in the GUI as a returned parameter of send http request function, is decoded and passed to the text area in correspondence with what command was executed.

Both servers tested on the AWS instance



Logger levels:

- severe is used when program crashes due to an error,
- warning is applied to cases when a program is still working, but desired functioning is deviated,
- info is for informative messages, like connection established/ended,
- fine, finer, finest – are informative and not at all urgent logs.

Part 4

Simple Messaging Architecture is created to allow developers focus on functionality other than on connection establishment and threads. By creating analogous classes to `DateClient` and `DateService`, it is possible to send requests to a server, execute them and send back responses to the client. The `Send` function on the server side should be used to execute and form the response. Message and port numbers should be updated and correlated in the newly created classes.

Server tested on AWS instance:

```
ubuntu@ip-172-31-3-115: ~/jar-files
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

12 packages can be updated.
0 updates are security updates.

New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

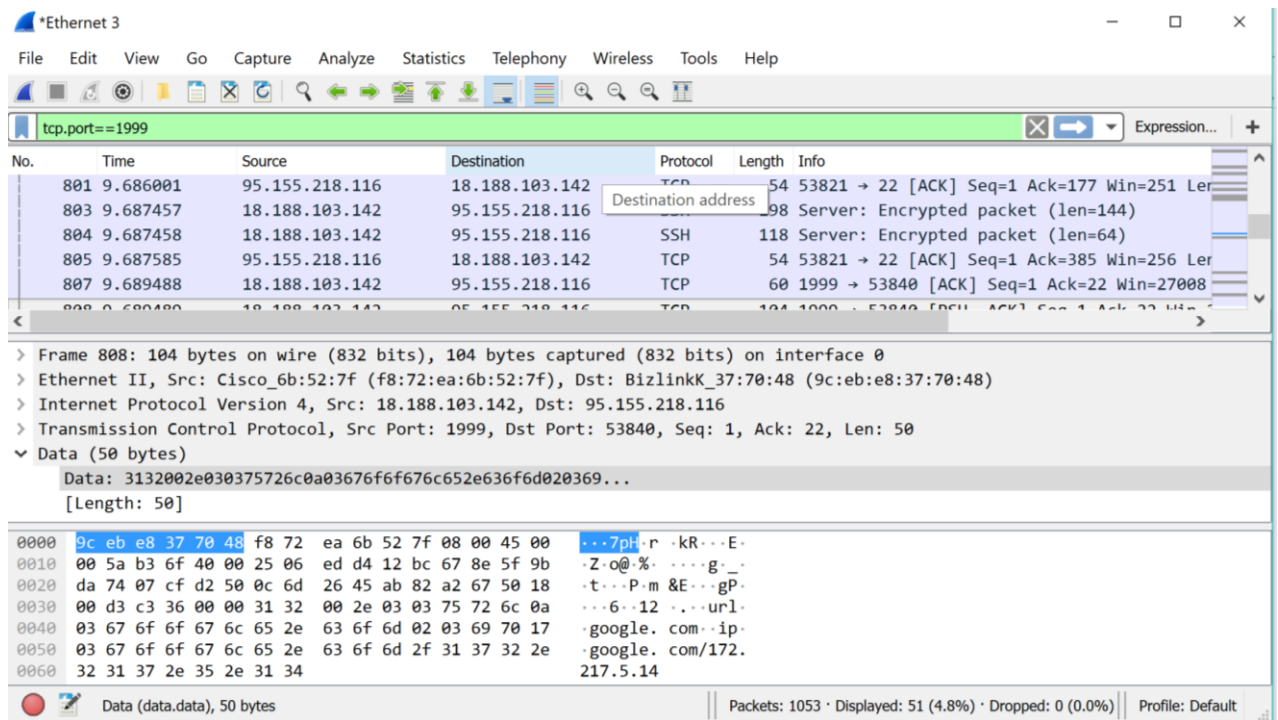
Last login: Sat Sep 22 11:37:18 2018 from 95.155.218.116
ubuntu@ip-172-31-3-115:~$ cd jar-files/
ubuntu@ip-172-31-3-115:~/jar-files$ java -jar SMAServiceDNS_lab1part4.jar
MessageServer: Simple Messaging Architecture (SMA) version 1.0
MessageServer: Created MessageServer instance fully!
MessageServer: MessageServer thread started. run() dispatched.
MessageServerDispatcher: Beginning of dispatch run() method.
MessageServerDispatcher: Received Message Message: type = 50 param = {url=google.com}.
MessageServerDispatcher: Received Message Message: type = 0 param = {$disconnect=$disconnect}.
MessageServerDispatcher: $disconnect found in Message Message: type = 0 param = {$disconnect=$disconnect}
-> Disconnect
MessageServerDispatcher: Beginning of dispatch run() method.
MessageServerDispatcher: Received Message Message: type = 50 param = {url=google.com}.
MessageServerDispatcher: Received Message Message: type = 0 param = {$disconnect=$disconnect}.
MessageServerDispatcher: $disconnect found in Message Message: type = 0 param = {$disconnect=$disconnect}
-> Disconnect
```

```
C:\Users\aaagri\elclipse-nw-progr\NetworkProgramming-1-gui-server-client\src>java -jar
SMAClientDNS_lab1part4.jar 18.188.103.142 1999 google.com
Hostname: google.com
IP address: google.com/172.217.5.14

C:\Users\aaagri\elclipse-nw-progr\NetworkProgramming-1-gui-server-client\src>
```

Wireshark capture

The reply message contained two parameter-value pairs: {"url"; <hostname>} and {"ip"; <hostname/ip>}.



Part 5

The concept: server has an SSL private key, which are integrated in the keystore. The client should create a public certificate and integrate it into the truststore locally. A key, a keystore and a truststore have their own different passwords. Truststore keeps trusted certificates, keystore has a private key to decrypt messages incoming with a public certificate.

InstallCert generates a certificate and puts it into the truststore called jssecacert – JSSE Certified Authority Certificate. While running InstallCert you can pass arguments in the following format:

host[:port] [password]

host – localhost or AWS IP, here we use AWS IP

port – we use 7070, but default 443 can be used. In the case of default port, https traffic should be allowed on port 443 in the security group.

password – specify it if you wish to use your own password. The password is *changeit* by default.

