Operating Systems 2 - Laboratory 3

Sockets and select

Write a **single-threaded program** called **`udpfwd`** which is responsible for forwarding incoming UDP datagrams. The program is configured via administrative TCP protocol. The program takes one positional argument: TCP port the program will listen on for incoming administrative connections.

The main process creates a TCP socket bound to localhost address port passed as an argument. Therefore, it won't accept connections from different hosts. At most 3 connections are allowed – if 4th client would like to establish connection the server should reply with an informative message and drop the connection.

The server simultaneously handles **one line string configuration commands** received from all clients along with UDP forwarding according to configuration. By default it forwards nothing (so it has no UPD sockets at the very beginning). Available commands are:

Command syntax and example	Stage	Description
`fwd <port`$_L$`> <IP`$_1$`:port`$_1$`> … <IP`$_n$`:port`$_n$`>` `> fwd 1500 127.0.0.1:2000`	4	Opens a new UDP socket and starts forwarding datagrams incoming on UDP port to all endpoints given in IP:port list. Up to 10 forwarding rules are allowed.
`close <port>` `> close`	5	Stops forwarding on given port and closes corresponding UDP socket.
`show` `> show`	6	Sends to a client an active configuration (open UDP ports with forwarding rules and filtering).

Keep in mind that command handling is not required in the laboratory part. While it is not required by the task you my find it useful to implement a simple UDP sending program during solving the homework part.

STAGES (TOTAL OF 14 POINTS)

	Stage	Points	Requirements
Laboratory part 105minutes	1	2	The program parses port argument, accepts a single TCP connection sends "Hello" message, drops the connection and exits. The must not accept connections from addresses other than localhost.
	2	3	The program accepts up to 3 TCP connections. Sends "Hello message" but does not drop client thereafter. It waits until client closes the connection an then is able to accept next one. Clients trying to connect over the limit are replied with informative message.
	3	2	Termination handling – after receiving SIGINT the server closes all connections and the listening socket and exits.
Homework 30.04.2021	4	3	The program parses fwd commands and correctly handles them. It should reject commands after applying 10 rules.
	5	2	The program parses and handles close command as described.
	6	2	The show command is working as described.

UPLOAD

Please upload your solution to: **/home2/samba/sobotkap/unix/**

You have to upload each stage immediately after finishing it. You upload to a network share via ssh.mini.pw.edu.pl server:

```
scp user.etapX.tar.bz2 user@ssh.mini.pw.edu.pl:/home2/samba/sobotkap/unix/
```

Please name your stages files according to the schema: `LOGIN.etapN.tar.bz2(.gz)`

THE STATEMENT

By decree 27/2020 of University Rector you must add the following statement to the uploads:

```
--------------------------------------------------------------------------
I declare that this piece of work which is the basis for recognition of
achieving learning outcomes in the OPS course was completed on my own.
[First and last name] [Student record book number (Student ID number)]
--------------------------------------------------------------------------
```

Please add it as comment at the beginning of each source file you upload.
Replace square brackets with your data.