Distributed Programming

Network Programming Test

To be submitted by June 28 2012, 23.59 at http://pad.polito.it:8080/ in the "Assignments" section

Develop an application that works like a simple TCP proxy, according to the following specifications.

1. At startup, the TCP proxy reads the following 3 arguments from the command line, **exactly in the order specified here**: the port number where the proxy listens to connection requests, the IP address and the port number of the *target server*, i.e. the TCP server the proxy will connect to.
2. The proxy behaves as a TCP server, listening to the port specified as first argument on the command line. When a connection has been established with a client, the proxy opens a corresponding connection with the target server and forwards data between the two connections (data coming from the client is forwarded to the target server and vice versa).
3. As soon as one of the connections is closed, a corresponding close must be performed by the proxy on the other corresponding connection.
4. In order to test the proxy, you can use the file-transfer client-server application developed in the Lab: first, start the file-transfer server, then start the proxy passing the IP and port number of the file-transfer server, and finally run the file-transfer client. The file-transfer application should work in the same way with or without the proxy.
5. The proxy must handle concurrent requests from several clients at the same time. There is no limit on the number of clients that can connect simultaneously to the proxy. As a consequence, for testing use any of the file-transfer server applications developed in the Lab that can handle concurrent requests.
6. The proxy application must be robust and must avoid memory leakages.
7. The C files of the proxy must all be included in a single zip archive created with this bash command:

   ```
   zip socket.zip *.c *.h
   ```

   Do not include the client and server files used to test the proxy, but include all the files that are necessary to compile the proxy application (it is possible to use files from the book by Stevens, but these files need to be included).

Warning: the submission system is *automatic.* Submission will be closed automatically at the deadline. Submitting the solutions in the last minutes is strongly discouraged.

Note: submissions will be considered valid only if it is possible to compile the application by running the following command:

```
gcc -o socket *.c -lpthread -lm
```

In case of doubts and questions about this assignment, first check the forum pages in the didattica.polito.it course website to see if somebody else already asked your question, otherwise use the forum (not email to teachers) to post your question so that the answer is available for everybody.