Distributed Programming

Network Programming Test

To be submitted by February 13 2014, 23.59

Develop a client-server application based on the C socket API and TCP connections, according to the following specifications:

1. The server receives exactly one argument on the command line: the TCP port number to which it listens.
2. The server must be able to serve concurrently several clients (at least 5, but no more than 20).
3. At startup, the server reads a binary file named *key.dat* in its local directory and stores its contents in memory. Let us denote k_0, k_1, ..., k_N the bytes of this file. If the file is not found, or if it is too large to be stored in memory, the server must exit immediately with an error message.
4. When connected to a client, the server receives a stream of bytes C=c_0, c_1, ... from the client and sends back to the client another stream of bytes R=r_0, r_1, ..., computed from C using the following formula:

   $$r_i = c_i \oplus k_{i \; mod \; N}$$

   where $\oplus$ is the bitwise exclusive-OR operator.
5. The connection must be closed gracefully in the following way:
   a. When the client terminates sending the input stream C, the client closes the client-to-server side of the connection.
   b. When the server detects that the client has closed the client-to-server side of the connection, the server finishes sending the remaining bytes and then closes the other side of the connection.
6. Client and server must be able to work with input streams of any length.
7. The client receives exactly three command line arguments, exactly in this order: the server name or IP address in dotted decimal notation, the server port number (as decimal number), and the name of a local file. The client must send the contents of this file to the server and store the response received from the server in another file, with a name computed as the concatenation of the original file name and the suffix "_rec". For example, if the file name is "test.dat", the file with the server response must be "test.dat_rec".
8. The client must try to keep the server as busy as possible, by applying this policy: the client must keep sending bytes whenever possible and must receive bytes with lower priority (i.e. only when sending more bytes is impossible).
9. Client and server must be developed for the Linux OS.
10. Design decisions about the implementation of the application are free but must be reasonable.

The C files of the client program must be saved under the directory `client`, the C files of the server program must be saved under the directory `server`. All the source files of the client and of the server must be included in a single zip archive created with this bash command:

```
zip socket.zip client/*.c client/*.h server/*.c server/*.h
```

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

The zip file with your solution must be submitted online by the deadline indicated above using the submission form available from inside the Politecnico campus at https://pad.polito.it/enginframe/dp1/dp1.xml or from outside the Politecnico campus at https://pad.polito.it:8080/enginframe/dp1/dp1.xml.

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 (from the directory where the archive has been extracted):

```
gcc -o socket_client client/*.c -I client -lpthread -lm

gcc -o socket_server server/*.c -I server -lpthread -lm
```

In case of doubts and questions about this assignment, first check the forum pages in the didattica.polito.it course web site 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.

Note that questions on the forum must regard exclusively clarifications about the specifications given in this text.

Do not ask questions about how to solve the assignment.