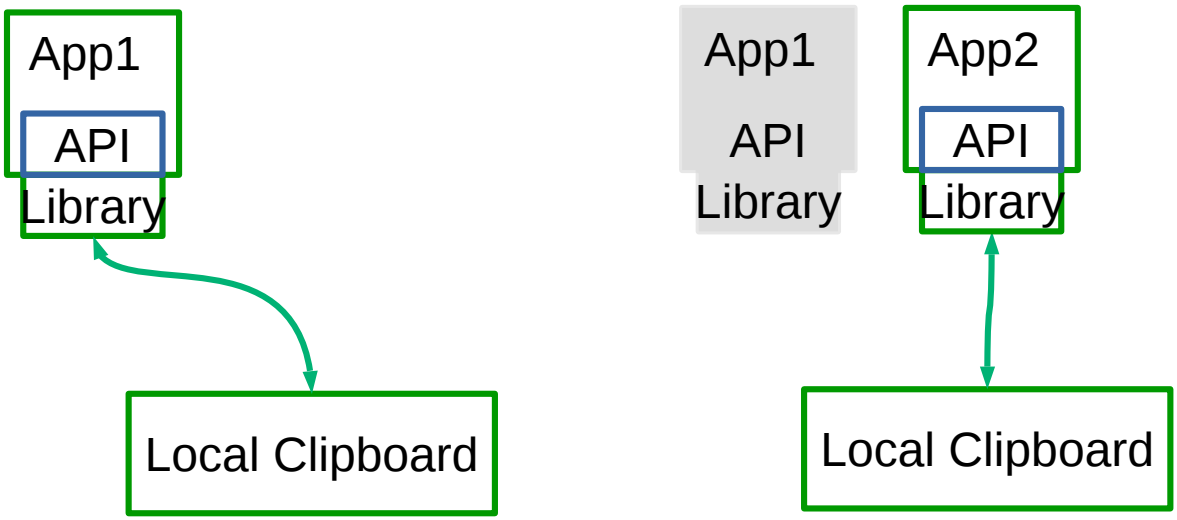
System Programming
6th Laboratory (10, 11 and 13 of April 2018)

The objective of this laboratory is to exercise communication using UNIX domain STREAM sockets and continue the implementation of the project

Objectives

in this laboratory students will modify the code developed in the previous laboratory by implementing the project API and the first prototype version of the server using UNIX domain STREAM sockets.
In this laboratory applications interacts with the server one at a time, and that communication is performed using one socket:



After the connect/access there is a direct connection between the client Application and the Local Clipboard to send requests and responses.

Clipboard

The clipboard will be a simple application that starts by creating on socket (UNIX domain and STREAM), assign it a well know address (to be defined in the .h file), do a listen and start an infinite loop that waits for the connection of a client and answer all its requests. All responses will be written to the same socket from where the requests are received.
The clipboard should have 10 regions as defined in the Project assignment.

API

Students should modify the following function described in the project assignment:

int clipboard_connect(char * clipboard_dir)

This function should try to do a connect to a server whose socket is created in the clipboard_dir. This function returns the file descriptor of the newly created and connected

socket or an error.

int clipboard_copy(int clipboard_id, int region, void *buf, size_t count)

This function receives the value returned by c**lipboard_connect** and uses it to send/copy data to clipboard, using a socket (all reads and writes will be done in the same file descriptor)

int clipboard_paste(int clipboard_id, int region, void *buf, size_t count)

This function receives the value returned by **clipboard_connect** and uses it to retrieve/paste data from the clipboard, using a socket (all reads and writes will be done in the same file descriptor)

API implementation

In order to ease the implementation of these three function it is necessary to define a suitable protocol.

In the **clipboard_copy** and **clipboard_paste** functions the application must send/receive a set of messages that should be formatted in a well defined form.
Both the application(API) and the clipboard should agree on the order and format of those messages.
To define them, students can use defined C structures.

Test application

The test application should read from the keyboard a string and an integer. The string will be copied to the clipboard (and later pasted) and the integer corresponds to the region.

Supplied code

Students should continue and change the code developed in the previous laboratory.

Error validation

When doing network communication a lot of additional errors can affect the client and the server.
Students should guarantee that all function that return an error code, get validated.
In case of error a suitable action should be taken (ignore, terminate processing, exit, …).
Students should study each function (in the man command) to know what are the return codes:

- socket
- bind
- listen
- connect
- accept
- send or write
- recv or read