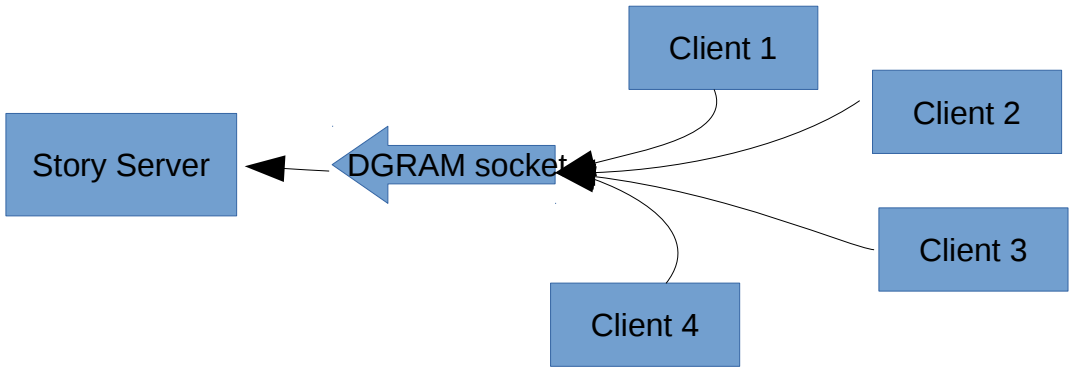
System Programming
6th Laboratory (5th and 7th April 2017)

This laboratory will exercise the use of Sockets to implement communication between clients and a server.

In this laboratory the students will implement one server (using UNIX domain sockets) that stores a story (concatenation of strings). The story is composed by the concatenation of strings sent by the various clients that connect to that servers. The overall architecture of the first version of the system is as follow:



Clients read from the keyboard a string and send it to the server.
The server, after reception of a messages concatenates it with the story (concatenation of the previously received messages). The server print of its screen the story stored in his memory.

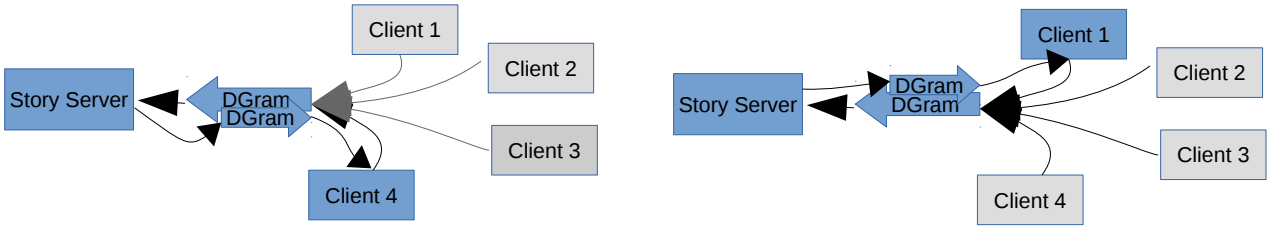
In order to stop the server the user should use the Ctr-C command. If the **`SIGINT`** signal is received the server should orderly exit.

1st version

In the first version of the server, the communication is one-way and performed using a Datagram socket (the server socket name in the **storyserver.h**).
The server should create the socket and bind an address.
Afterward it should loop waiting for new messages.
In the first version of the protocol, it is advisable to use the **message** structure (also defined in **storyserver.h** the should be used).

2nd version

In the second version of the system, the server should send back to the client the complete story, after reception of the message and its concatenation with the current story.
In order to make datagram sockets bidirectional it is necessary for the client to bind an address to the socket.



The clients socket name should follow the following structure **`./sock_cli_%d`** (where %d will be replaced by the PID of the process.
If the story is larger than message buffer, the client should be notified of that.

3rd version

The third version of the protocol will use stream sockets.
The server, besides doing the bind, should also do a listen, followed by several accepts.
The client should do a connect.
Right after handling the client request, the newly created socket (returned by the accept) should be closed and the server should return to the accept.
What happens if a client does not send any message?
The server should send all the story (independently of its size to the story).

4th version (INET)

Modify the 3rd version of the server to use Internet domain sockets.
The server should create an AF_INET socket, bind an address (INADDR_ANY and port defined in the .h file).

API information

- http://www.gnu.org/software/libc/manual/html_node/Sockets.html
- http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html
- http://pubs.opengroup.org/onlinepubs/9699919799/functions/V2_chap02.html#tag_15_10
- http://beej.us/guide/bgnet/
- http://tldp.org/LDP/lpg/node7.html

SOCKETS

- man socket
- man 2 bind
- man 2 accept
- man 2 connect
- man unix
- man 7 signal