Lab 9 - FIFO

Use FIFO to write a client side of program to work with the server side program *server.c*. This client/server application performs a simple instant messaging between two users on the same machine.

The server program is started first, waiting for client to connect. The user on the client side can decide to quit the conversation by typing CTRL-C. When the client stops the conversation, only the client program is terminated. The server program will wait for the next connection from client. The server will terminate if CTRL-C is received from the user. When the server is terminated, the client is also terminated.

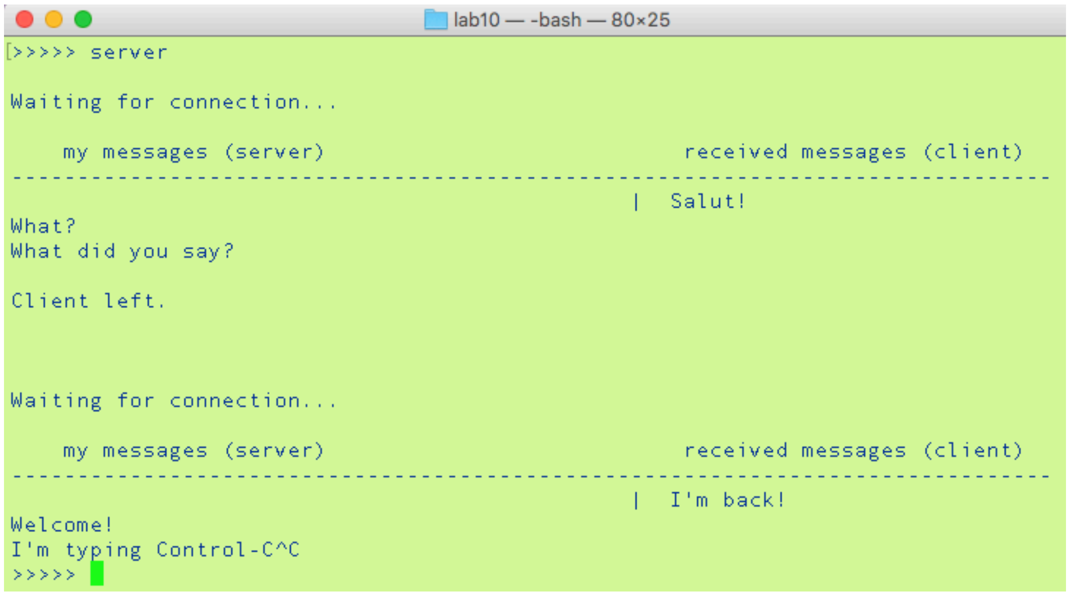
The communication should be done using two FIFOs. They are created by the server using /tmp.

Part of the *server.c* code is provided. Note that the dotted lines represent some missing code.

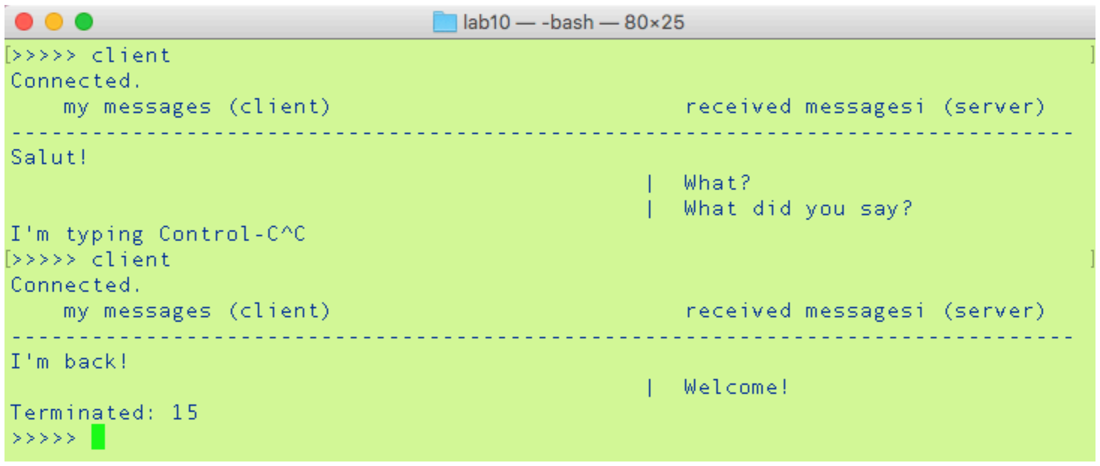
Please replace the string *name* in /tmp/*name*0 and /tmp/*name*1 in the code by your own name.

You need to have a partner to test each other's solution in the laboratory using a CS server.

Sample run:



```
>>>>> server

Waiting for connection...

    my messages (server)                              received messages (client)
--------------------------------------------------------------------------------
                                               |   Salut!
What?
What did you say?

Client left.



Waiting for connection...

    my messages (server)                              received messages (client)
--------------------------------------------------------------------------------
                                               |   I'm back!
Welcome!
I'm typing Control-C^C
>>>>>
```



```
>>>>> client
Connected.
    my messages (client)                           received messagesi (server)
--------------------------------------------------------------------------------
Salut!
                                               |  What?
                                               |  What did you say?
I'm typing Control-C^C
>>>>> client
Connected.
    my messages (client)                           received messagesi (server)
--------------------------------------------------------------------------------
I'm back!
                                               |  Welcome!
Terminated: 15
>>>>> 
```

server.c

```
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <sys/stat.h>

int main(int argc, char *argv[]) {

    int fd1, fd2;
    pid_t pid;
    char ch;
    char blanks[55]="                                              |  ";
    int  writeblanks = 1;

    unlink("/tmp/name0");
    unlink("/tmp/name1");
    if ( mkfifo("/tmp/name0", 0777) || mkfifo("/tmp/name1", 0777)) {
        perror("fifo");
        exit(1);
    }
    chmod("/tmp/name0", 0777);
    chmod("/tmp/name1", 0777);
    while (1) {
        printf("\nWaiting for connection...\n");
        fd1 = open("/tmp/name0", O_RDONLY);
        fd2 = open("/tmp/name1", O_WRONLY);
        printf("\n    my messages (server)                    received messages (client) \n");
        printf("--------------------------------------------------------------------------------\n");
        if ( (pid = fork()) == -1 ) {
            perror("fork");
            exit(1);
        }
```

```
            if ( pid == 0 )
                  while (1) {
                        ……..
                  }
            while ( read(fd1, &ch, 1) == 1 ) {
                  if ( writeblanks == 1 )
                        write(1, blanks, sizeof(blanks));
                  write(1, &ch, 1);
                  ……..
            }
            close(fd1);
            close(fd2);
            printf("\nClient left.\n\n\n");
            kill(pid, SIGTERM);
      }
}
```