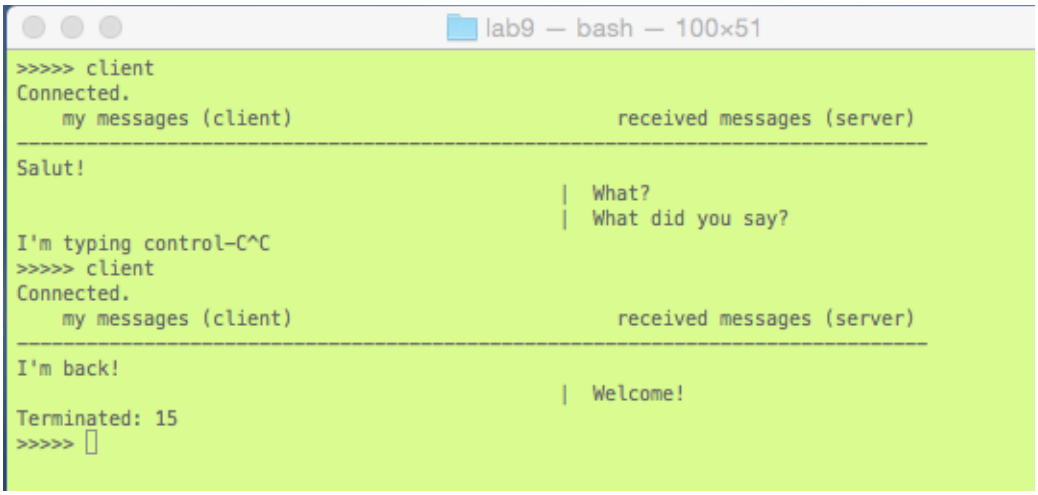
Lab 10 - FIFO

Use FIFO to write a client side of program to work with the given 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 in the current directory.

Sample run:



server.c

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

void stdinread(int fd);

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

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

	unlink("./fifo1");
	unlink("./fifo2");
	if ( mkfifo("./fifo1", 0777) || mkfifo("./fifo2", 0777)) {
		perror("fifo");
		exit(1);
	}
	while (1) {
		printf("\nWaiting for connection...\n");
		fd1 = open("./fifo1", O_RDONLY);
		fd2 = open("./fifo2", 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) {
                read(STDIN_FILENO, &ch, 1);
                if ( write(fd2, &ch, 1) == -1 )
                    exit(0);
            }
        while ( read(fd1, &ch, 1) == 1 ) {
            if ( writeblanks == 1 )
                write(1, blanks, sizeof(blanks));
            write(1, &ch, 1);
            if ( ch == '\n' )
                writeblanks = 1;
            else
                writeblanks = 0;
        }
        close(fd1);
        close(fd2);
        printf("\nClient left.\n\n\n");
        kill(pid, SIGTERM);
    }
}
```