Remote Shell for UNIX

This project was made for Distributed Programming Course. This course was taught by Professor Felicia Ionescu at students from Thoroughgoing Studies in Engineering Computing Systems at University "Politehnica" Bucharest, Faculty of Electronics and Telecommunications.

This project had been my first distributed project and because of that it is a limited project.

This RShell allow us to execute commands on target machine, this commands are BASH commands, redirecting the result on the host machine. The host machine is the computer who lunch the client and target machine is the computer who lunch the server. It is a much simple program then Telnet because it has the following limitations:

- because it doesn't support login and su we can understand that the connected user have the rights of the user which have lunched the server on the target machine.
- it doesn't support programs who works in graphical level. (eg: we can't lunch a xterm because it will work on target machine and we will lose control on him)
- it doesn't support getting files because it wasn't thought to have this feature.
- the server support only one client because the main program wait for his child.
- the client support two pagers: less and more which are the most common pagers in UNIX systems.

The program is implemented in ANSI C for UNIX, so it has the maximum portability. The only request is that computer to have a compiler compatible gcc, BASH and one of the following pagers: less or more.

The communication protocol was chosen to be TCP/IP because it have confirmation so we don't need to have a broadcast which is the feature of UDP. Over the TCP it is the application protocol:

- it send a packet of 515 characters divided in two blocks: one of 512 characters contain the data and second of 3 characters who are for synchronization.
- the synchronization characters are always the last three characters from the packet.
- server send the "000" if it isn't the last packet from message and "END" if it is the last packet form message.
- client don't send synchronization characters.
- for exit we send "exit" in data area.
- communications are made on port 12345.