

Tarea 1 - Cliente multi-telnet

Profesor: Catalina Álvarez

Auxiliar: Ivana Bachmann

Objetivos

En esta tarea deberán implementar un cliente telnet que permita interactuar con varios servidores simultáneamente; la idea es que tengan varios servidores telnet corriendo y su cliente permita enviar comandos a uno de ellos o a todos al mismo tiempo.

Se les provee con la implementación de un servidor Telnet de ejemplo que pueden usar para sus pruebas, cuyos comandos se explican más adelante.

Implementación pedida

Vamos a listar las características más importantes de lo que deben implementar:

- La implementación básica es un cliente Telnet:
 1. Una aplicación que reciba el nombre de un server y un puerto (por defecto el 23) y abra un socket a dicho server.
 2. Se debe estar esperando información desde el usuario (comandos) y respuestas desde el servidor simultáneamente.
 3. Los comandos ingresados por el usuario se dividen por salto de línea, y se envían sin procesamiento alguno al server.
 4. El server recibe los comandos y los procesa. El resultado del procesamiento se envía directamente al cliente.
 5. El cliente muestra el resultado en pantalla.
- Su cliente multi-telnet debe tener la misma funcionalidad anterior, pero con varios servidores a la vez.
 1. Los servidores con los cuales deben operar pueden entregarse en dos formatos (ustedes deben implementar ambos):

- a) Por línea de comando: Al llamar a su cliente, pueden incluir los servidores como argumentos en el siguiente formato: *nombre dirección puerto ...*
 - b) Usando un json: Al llamar al cliente, pueden darle como parámetro un json que contiene la misma información anterior en el formato: "nombre": ..., "dirección":..., "puerto":...
2. Los servidores deben distinguirse por el nombre que se les da al ser creados. En caso de dar nombre duplicados, quedan en libertad de acción (deben dejar documentada su elección).
3. Al igual que el caso simple, el cliente debe estar escuchando resultados desde los servidores y comandos desde el usuario. Eso sí, ahora se deben distinguir los resultados usando el nombre o dirección del server del cual viene; queda a su criterio como hacerlo. Además, cuando reciban resultados de varios servidores a la vez, no deben mostrarlos mezclados, sino esperar tener resultados completos antes de entregarlos al usuario.
4. Finalmente, su cliente debe permitir interactuar con uno/varios/todos los servidores a la vez. Es decir, deben permitir los comandos:
 - a) `ls name1`
 - b) `ls name1 name2`
 - c) `ls all`

Servidor provisto

Para facilitar su trabajo, se les provee un servidor Telnet simple que pueden correr varias veces para realizar sus pruebas. El servidor provee las siguientes funcionalidades:

1. Cuando se levanta el servidor se le puede entregar como argumento el puerto a través del cual trabaja (por defecto es el 8000).
2. Responde a llamadas simples al sistema operativo: *ls*, *cat* y *echo*, además de *help* y *exit*.

El servidor se les entrega como un binario compilado en Linux (**server**) junto a las librerías que requiere para correr. Por tal razón, el servidor solamente corre en Linux (además de que las llamadas al sistema operativo son exclusivas de Unix).

Informe

Aparte del código fuente de su tarea, **deben entregar un README explicando**

como correr su tarea, y aclarando sus supuestos. Por favor consideren que mientras mejor documentada su tarea, más fácil su evaluación y menos problemas tendrán.

Implementación

Ustedes deben implementar lo pedido en uno de los siguientes lenguajes de programación: **Java**, **C**, **C++** o **Python**. En cualquiera de los casos, deben consultar al auxiliar o profesor en caso de que deseen usar alguna librería/framework fuera del core del lenguaje.

La posibilidad de utilizar otros lenguajes no está cerrada, pero debe conversarse con la auxiliar previamente (básicamente para asegurarse de que su tarea sea revisable).

Cualquier duda o pregunta o reporte de bugs, dirigirse al foro de U-cursos.

Evaluación

Esta tarea será evaluada tomando en cuenta la funcionalidad de su cliente (cumple con las funcionalidades) y la facilidad/dificultad que tenga el corrector de hacerla funcionar.