

Taller de IPC

Sistemas Operativos

08 de Septiembre de 2015

1. Mini Servidor Telnet

1.1. Parte 1

Se tiene el servidor de MINI-TELNET provisto por la cátedra con las siguientes características:

- Recibe mensajes UDP. Puerto 5555. Longitud máxima 1024.
- Ejecuta el comando recibido.
- El output de dicho comando **no se envía de vuelta al cliente**.

Implementar el cliente de MINI-TELNET de forma tal que:

- Debe recibir como parámetro la IP del servidor.
- Envía mensajes hasta que alguno es “chau” y termina.

1.2. Parte 2

Modificar el servidor de la parte 1 para que tenga las siguientes características:

- Recibe mensajes TCP. Puerto 5555. Longitud máxima 1024.
- Ejecuta el comando recibido.
- El output de dicho comando **se envía de vuelta al cliente**. Tanto `stdout` como `stderr`.

Modificar el cliente de la parte 1 para que se conecte al nuevo servidor y tenga las siguientes características:

- Debe recibir como parámetro la IP del servidor.
- Debe mostrar el output de cada comando ejecutado en el servidor.
- Envía mensajes hasta que alguno es “chau” y termina.

2. Mini Shell

Se pide implementar parte de la funcionalidad de un de `shell` minimal. El mismo solo soporta comandos de dos formas: **a)** el nombre de programa (más argumentos), por ejemplo, `ls -al`; o **b)** más de un programa comunicados por `|`, por ejemplo, `ls -al | wc -l | awk '{ print $2 }'`.

- Implementar la función encargada de ejecutar, como un subprocesso, cada uno de los programas que forman el comando. La misma retorna 0 en caso de éxito y -1 en caso de error, y tiene la siguiente aridad:

```
int run(char *program_name[], char **program_argv[], unsigned int count)
```

`program_name`: Es un arreglo con los nombres de los programas del comando ingresado.

`program_argv`: Es un arreglo de punteros a los argumentos de cada programa invocado dentro del comando.

`count`: Es la cantidad de programas invocados en el comando.

Tener en cuenta que los subprocessos tienen que poder comunicarse entre sí de modo de que el comando `ls -al | wc -l | awk '{ print $2 }'` se comporte de la misma manera que lo haría en cualquier otro shell.

- ¿Por qué es importante cerrar los extremos de los `pipes` que no se utilizan?