Pontificia Universidad Católica

Sistemas Operativos y redes

Tarea 3

Alumnos:
Sebastian Butorovic
Gabriel della Maggiora

Profesor:
Cristian Ruz

17 de Noviembre de 2015

Índice

1. Compilación

Esta tarea fue programada en lenguaje C/C++ . La compilación de la tarea se debe hacer con el comando ***make*** en la carpeta src, el cual ejecuta la siguiente línea de comando:

g++ tarea3.cpp Server.cpp Client.cpp -o tarea3

Este comando generará un ejecutable llamado ***tarea3*** que sirve para iniciar tanto un servidor como un cliente mediante ./tarea3.

2. Uso Servidor y Cliente

Tanto el cliente como el servidor comparten el ejecutable, pero se llaman con distintos parametros. La forma de llamar a cada uno es:

Servidor: **./tarea3 s**

Cliente: **./tarea3 c IP port**

Esto es diferente a la forma de ejecutarlos del ejemplo del enunciado en donde el servidor se ejecuta con el comando **./servidor** y el cliente con **./cliente**

2.1. Servidor

Para inicializar una instancia de un servidor, primero es necesario definir el archivo fileserv.conf. Este archivo contiene los parámetros con los que el servidor debe funcionar. Este archivo debe tener dos lineas y su formato debe ser:

```
portNumber
directoryPath
```

Una vez definido el archivo fileserv.conf, se debe ejecutar

./tarea3 s

para iniciar una instancia de un servidor. Cabe destacar que la IP usada para iniciar el servidor es 127.0.0.1 y esta incluida dentro del código, ya que según el enunciado solo se debe entregar el numero de puerto y el directorio en el archivo

fileserv.conf, pero no la IP. Para cambiarla se debe cambiar el argumento pasado al constructor del servidor, en la función main del archivo tarea3.cpp y volver a compilar.

Al iniciar una instancia del servidor, se buscará en el directorio entregado en fileserv.conf un archivo llamado filePermissions.txt. Este archivo contiene un listado de todos los archivos junto a el usuario dueño y si está o no compartida. El formato de cada linea del archivo es la siguiente:

nombreArchivo;nombreDueño;shared/notshared

Si el programa no encuentra este archivo, lo creara y añadirá a el todos los archivos presentes en el directorio entregado, dejando como dueño de esos archivos al usuario de nombre 'Admin' y los dejará como "notshared".

2.2. Cliente

Para inicializar una instancia de un cliente, se debe ejecutar el comando:

./tarea3 c IP port

La primera acción que se debe realizar es la de identificarse como usuario, ya que el servidor no deja realizar comandos a menos que se esté identificado.

3. Comandos

Se implementaron los siguientes comandos cuyo uso se detalla a continuación:

3.1. User

En la interacción con el cliente su uso es el siguiente: user,<username>. Se detalla un ejemplo a continuación.

Figura 1: Login con comando user

3.2. Get

Para realizar el comando get, es necesario dos cosas, primero haberse identificado como algún usuario y segundo que el dueño del archivo sea el usuario o bien el archivo esté compartido. Su uso es el siguiente: get,<nombre de archivo en el repo>. Se detalla un ejemplo a continuación.

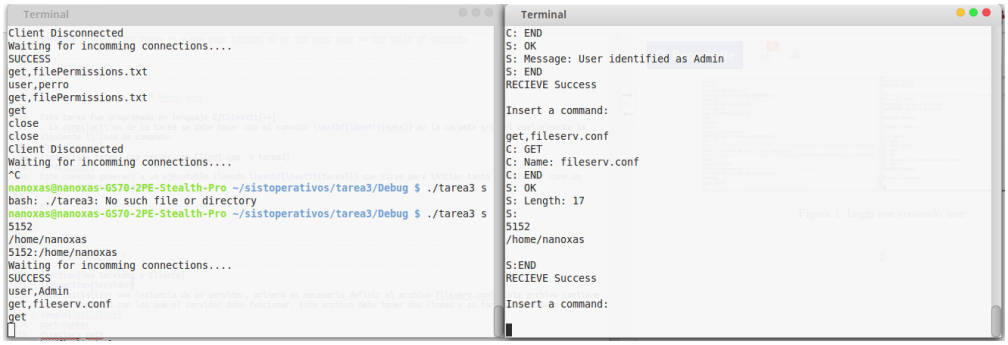

Figura 2: Obtención de archivo con get

3.3. Put

Para realizar el comando put, es necesario haberse identificado como algún usuario. Su uso es el siguiente: put,<path absoluto del archivo a poner el repo>. Al subir un archivo al repositorio se agregará una linea en el archivo filePermissions.txt indicando el nombre del archivo, su dueño y por defecto, que no esta compartido. Se detalla un ejemplo a continuación.

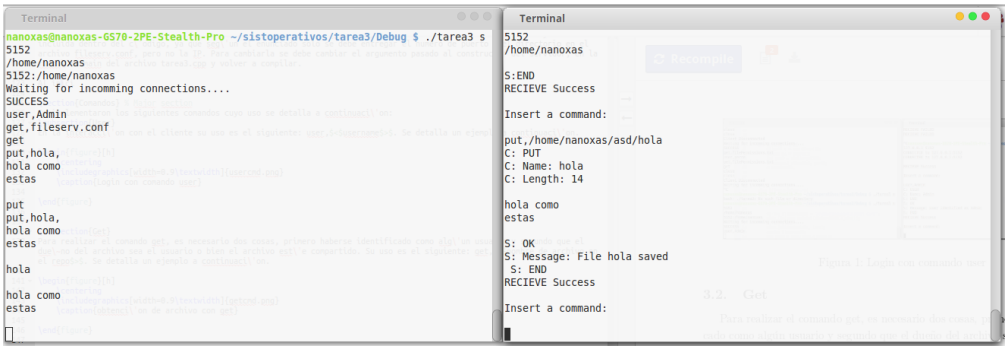

Figura 3: Subida de un archivo con put

3.4. Ls

Para realizar el comando ls, es necesario haberse identificado como algún usuario. Su uso es el siguiente: ls. Se detalla un ejemplo a continuación.

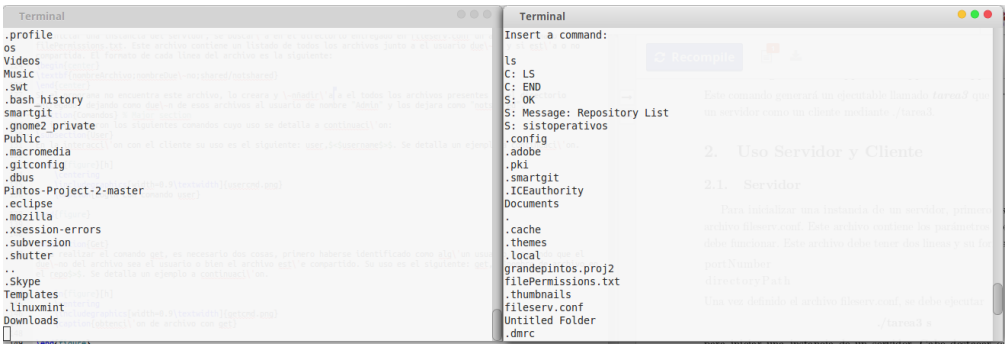

Figura 4: Listado del directorio con ls

3.5. Rm

Para realizar el comando rm, es necesario dos cosas, primero haberse identificado como algún usuario y segundo que el dueño del archivo sea el usuario. Al remover exitosamente un archivo del repositorio, se eliminara la linea correspondiente a ese archivo del archivo filePermissions.txt. Su uso es el siguiente: rm,<nombre de archivo en el repo>. Se detalla un ejemplo a continuación.

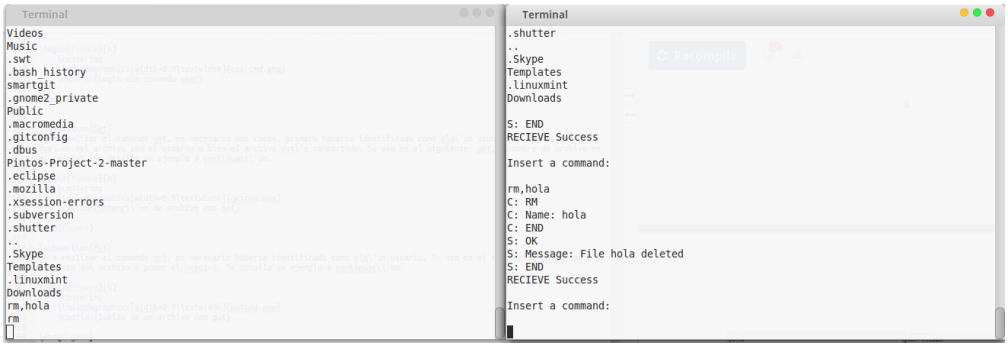

Figura 5: Eliminación de un archivo con rm

3.6. Share

Para realizar el comando share, es necesario dos cosas, primero haberse identificado como algún usuario y segundo que el dueño del archivo sea el usuario. Su uso es el siguiente: share,<nombre de archivo en el repo>. Este comando cambia el ultimo parametro de la linea correspondiente al programa en el archivo filePermissions.txt, dejandolo como shared. Se detalla un ejemplo a continuación.

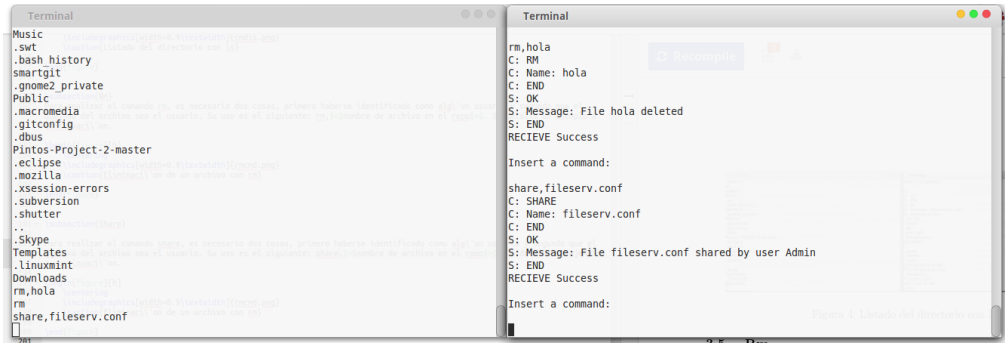

Figura 6: Compartir un archivo con share

3.7. Close

Para realizar el comando share, es necesario haberse identificado como algún usuario. Su uso es el siguiente: share. Se detalla un ejemplo a continuación.

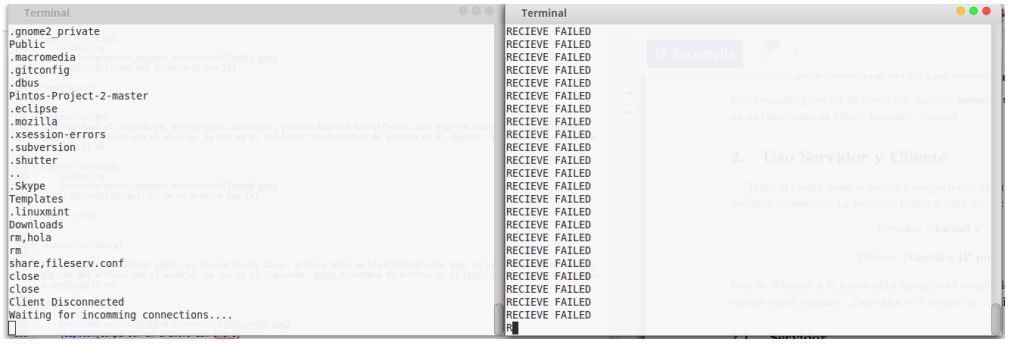

Figura 7: Cerrar la conexión