

Shell restringido

Javier Falcón (2016-5265)

1 Ejercicio 1

Este ejercicio consistió en la realización del comando *remove* para poder eliminar archivos desde el shell de **Pintos**. Antes de adentrarme en Pintos, fue necesario familiarizarme con la función de *getopt* existente en las librerías del lenguaje C. Por lo que creé un pequeño fichero fuente en C que utilizaba los *system calls* de Linux para realizar la eliminación. Este código serviría de estructura base para el fichero funcional que se implementaría adentro de Pintos. En vista de que Pintos tiene una librería muy limitada, habría que adaptar el código más adelante.

El fichero *rm.c* funciona de la siguiente manera:

Encontramos dos funciones principales → **remove** y **doRemove**. En éstas, se aloja el código que elimina archivos en el *filesystem* de Pintos.

La función **remove** hace el parseo del comando que le llega como argumento en un arreglo bidimensional y es la que se llama desde el código del shell. Luego del *parsing*, se procede a ejecutar la acción correspondiente al comando ingresado. En esta función se verifica adicionalmente si el comando ingresado tiene los argumentos necesarios para funcionar, y se hace la llamada interna a **doRemove** solo si el comando es válido y no se eligieron las opciones de **-help** **-version**.

Finalmente, la función **doRemove** se encargará de llamar a la función del sistema llamada **filesys_remove**, la cuál removerá el archivo que se le ingrese como parámetro.

2 Ejercicio 2

Este ejercicio consistió en la realización del comando *move* para poder mover o trasladar archivos desde el shell de **Pintos**.

Esta función fue la más compleja. Básicamente utiliza una lógica muy parecida al **remove** en cuanto a lo que sería el parseo del comando. El archivo *mv.c* contiene 2 funciones principales: **move** y **doMove**.

La función **move** es la que se encarga de realizar el parseo del comando y de ejecutar las acciones que se corresponden a cada opción que se invoque. La lógica es idéntica que la implementada en *rm.c*.

La parte interesante se encuentra en el **doMove**. Esta función es la que se encarga de trasladar el finchero solicitado a otro *path*. Una cosa importante es

que el código de **doMove** está fuertemente inspirado en la función **fsutil.cat**, la cual se encarga de abrir un archivo y mostrar el contenido en consola. La razón por la que se utilizó como guía esta función es debido a que tienen una lógica parecida, solo que en vez de escribir el contenido en consola, se escribe en un archivo nuevo de mismo nombre que fue creado en la dirección de destino. Para trasladar un archivo primero se abre el *file* que se va a mover con el objetivo de acceder a sus contenidos y se prosigue a construir el nuevo *path*, que será el sitio donde se moverá el archivo. Luego, se abre el nuevo archivo en la dirección de destino, de tal manera que quede preparado para que escriban sobre él. En tercer lugar, utilizando un *stream*, se copia el contenido del archivo original al nuevo, que fue creado en el nuevo alojamiento. Finalmente, se borra el archivo original para quitar su rastro de allí.