

# Shell Restringido

---

Nombre: Miguel Estevez

Matricula: 20170200

## Ejercicio 1. <rm>

Para la implementación del comando *rm* se utilizó la interfaz *filesys* para el manejo de archivos de sistemas. Lo primero que se modificó fue el *make* de *threads* ya que el *init* y el *shel.c* se encuentran en esa carpeta, se definió la directiva "FILESYS", y para compilar también se incluyó el directorio */filesys*. Para poder hacer uso de esta interfaz se debió haber creado un disco simulado con una partición del sistema de archivos. Esto se logró con los siguientes comandos:

```
pintos -mkdisk filesys.dsk --filesys -size = 2
pintos -f -q
```

Lo que hacen es primero crear un disco simulado "filesys.dsk" de 2 MB y a continuación se formatea. Una vez que el disco de *pintos* está listo para manejar los archivos se procedió a la implementación del comando *rm*.

Este comando hace uso de la función de *filesys.c* *filesys\_remove* llamándola con el nombre de cada archivo que se requiere eliminar. Al esta función devolver *true* si se eliminó con éxito (indicando que existe) se aprovecha y se imprime que la acción se realizó si la opción *--verbose*, *-v* fue especificada.

De lo contrario se imprime en pantalla que el archivo es inexistente a no ser que se haya especificado la opción *-f*, *--force* que evita esta acción.

Las opciones *--help*, *--version* siguen su comportamiento natural e imprimen en pantalla ayuda o la versión respectivamente. Ambas tienen la máxima prioridad pero se ejecuta la que más a la izquierda esté.

La implementación de éste comando se hizo en el archivo *rm.c*, cuya función principal es *rmProc*. El flujo de ejecución de la misma es el siguiente, se clasifican los token que comienzan con '-', y los otros se guardan en un arreglo ya que esto son clasificados como nombre de archivos. Al finalizar de analizar todos los token, se itera sobre el arreglo que contiene los nombres de los archivos y llamando la función *filesys\_remove* son removidos.

La documentación se encuentra en: [html/rm\\_8c.html](html/rm_8c.html)

## Ejercicio 2. < mv >

Este comando al igual que el anterior hace uso de la interfaz *filesys*, pero en este caso además de usar funciones de *filesys.c*, hace uso de funciones de *file.c*. Ambas del directorio *filesys*. La implementación de este comando se hizo en el archivo *mv.c*. Su función principal es *mvProc()*. El flujo de la misma es el siguiente.

Primero se buscan todos los token que sean opciones (aquellos que comienzan por '-') y se van marcando en flags, al igual que en rm las opciones --help, --version siguen su comportamiento natural. Como éstas tienen la máxima prioridad la primera que se detecte imprime en pantalla lo solicitado y hace terminar el flujo de la función.

A continuación, se procede con la ejecución normal, se verifica el número de operandos, en caso de error se imprime en pantalla y se termina la función, de lo contrario se intenta abrir el archivo de origen mediante la función `filesys_open` que devuelve un `struct file`, para comprobar su existencia en caso de error se notifica y no se continúa.

Se procede a comprobar la existencia del archivo de destino de la misma manera, si existe y la bandera de `-n`, `--no-clobber` está prendida entonces se culmina el proceso, en caso contrario se elimina dicho archivo para poder crearlo con el tamaño que concuerde con el del archivo de origen.

Se verifica la existencia de data en el archivo de origen para su transferencia hacia el archivo de destino. Esto se logra cargando toda la data en un buffer de memoria que previamente se reservó con `malloc`, para cargarla se usa la función `file_read` que lee del archivo especificado por parámetro en seguida se escribe en el archivo destino con `file_write` y se libera la memoria del buffer.

Finalmente se elimina el archivo de origen y si fue especificada la opción `--verbose`, `-v` se imprime en pantalla la descripción de la acción que se ejecutó.

La documentación se encuentra en: [html/mv\\_8c.html](http://html/mv_8c.html)