

# Sistemas Operativos

Ingeniería Técnica de Sistemas 2011

## Práctica #4 (Mini Sistema de Ficheros (mfs))

### Ejercicio 1 (Parte obligatoria)

- Implementar las funciones suficientes para que **mfs\_get**, **mfs\_put**, **mfs\_cp** y **mfs\_ls** funcionen.
- Escriba la función **mfs\_unlink()** para que se puedan borrar ficheros.
- Escriba la función **mfs\_rename()** para que se puedan renombrar ficheros.
- Escriba la función **mfs\_mkdir()** y **mfs\_rmdir()** para que se puedan crear/borrar subdirectorios. Modifiquense también las otras funciones que lo requieran.
- Tiene que funcionar la asignación de tamaño con extents.

La asignación de espacio a los ficheros se realizará con extents. Inicialmente puede suponerse que solo existirá un extent. Para las entradas en directorio, es suficiente con usar una tabla con nombres de tamaño fijo. Usar un inodo por bloque. La gestión de tamaño libre se realizará con un bitmap, que estara colocada entre el superbloque y la tabla de inodos. El sistema de ficheros se usará como fichero `my_mfs.img`. Notese que todos los programas estan implementados usando el sistema de ficheros normal como ejemplo, para que se sepa cual seria su uso.

### Ejercicio 2 (Inodos)

Permitir más de un inodo por bloque.

### Ejercicio 3 (Implementa opcion -l al comando mfs\_ls)

Escriba la función **mfs\_stat()** para que se puedan conseguir información adicional de cada fichero.

### Ejercicio 4 (Links (mfs\_ln))

Escriba la función **mfs\_link()** para que se le pueda dar más de un nombre a un fichero. Modificar **mfs\_unlink()** de forma acorde.

### Ejercicio 5 (Subdirectorios (2))

Modifique **mfs\_cp()** y **mfs\_mv()** para que acepten argumentos de la forma: **programa <lista ficheros>directorio**, para poder trabajar con una lista de ficheros cuando el destino es un directorio.

### Ejercicio 6 (Directorios)

Usar tamaño variable para cada entrada en el directorio.

### Ejercicio 7 (Información de debug)

Escribir un programa **mfs\_debug** que dado un sistema de ficheros, escriba por pantalla superbloque, inodos y bloques, con información relevante.

**Ejercicio 8 (Crear un comprobador del sistema de ficheros)**

A partir de `mfs_debug()` crear un reparador/comprobador del sistema de ficheros. Ayudar su necesidad de uso marcando en el superbloque si el sistema de ficheros esta limpio o no.

**Ejercicio 9 (Poder cambiar el nombre del sistema de ficheros)**

Permitir cambiar el nombre del sistema de ficheros. Para ello usar una variable de entorno (`MFS_NAME`). Ver la función `getenv()` del sistema para recuperar variables de entorno.