

Ejercicios Ficheros ASO

Usar las funciones disponibles en el API del sistema en los siguientes casos. En cada ejercicio añadir los ficheros de cabecera (`#include`) necesarios para que no se produzcan avisos en la compilación.

Ejercicio 1. Comando `mys`

Escribir un programa con una funcionalidad semejante a la del comando `ls` que cumpla con las siguientes especificaciones:

El programa admite dos posibles opciones mutuamente excluyentes:

- **opción -d:** `-d ruta_directorio`
- **opción -f:** `-f ruta_fichero_regular`

Con la opción `-d` el programa debe comprobar que el argumento que se le ha pasado es un directorio sobre el que se tienen permisos de lectura y ejecución y caso de que sea así deberá recorrer las entradas de dicho directorio mostrando:

1. Si la entrada se corresponde con un fichero regular escribirá el nombre de dicho fichero y entre paréntesis el tamaño en kbytes de dicho fichero y el numero de enlaces rígidos. Si dicho fichero tiene permisos de ejecución se escribirá un `*` despues del paréntesis. Ejemplo:

```
unfichero.txt (1.5 kb, 1 link)
unejecutable (3 kb, 1 link)*
```

2. Si la entrada se corresponde con un directorio escribirá el nombre de dicho directorio entre corchetes y entre parentesis el número de enlaces. Ejemplo:
[undirectorio] (2 links)

3. Si la entrada se corresponde con un un enlace simbólico escribirá el nombre de dicho enlace y a continuación, entre paréntesis el nombre del fichero al que apunta. Ejemplo:
unenlacesim (/home/manuel/unfichero2.txt)

Con la opción `-f` el programa debe comprobar que el argumento que se le ha pasado es un fichero regular y caso de que sea así deberá mostrar su nombre y entre paréntesis el número de inodo del fichero y su tamaño en kilobytes. Ejemplo:

```
ejfichero(inodo 18900, 1.5 kb)
```

Ejercicio 2. Cerrojos

a) Escribir un programa que consulte y muestre en la salida estándar el estado del cerrojo sobre un fichero que se pasa como argumento. El proceso mostrará el estado del cerrojo y teniendo en cuenta dicho estado:

- Si el cerrojo esta desbloqueado, se establecerá un cerrojo de escritura en el fichero y escribirá la hora actual. Después suspenderá su ejecución durante 40 segundos (`sleep`) y a continuación liberará el cerrojo.
- Si el cerrojo está bloqueado se terminará el proceso.

El programa no deberá modificar el contenido del fichero si el cerrojo esta bloqueado.

b) Comprobar si es posible modificar un fichero que ha sido bloqueado con un cerrojo de escritura.

