

Práctica 2

Desarrollo extra

2. Desarrollo extra	1
2.1. Descripción	1
2.2. Desarrollo de la parte opcional	2

2.1. Descripción

Implemente la operación **creación de enlaces rígidos: ln** e inclúyala en la estructura **fuse_operations**. Consulte el manual de FUSE para ver el prototipo de la función `link`¹

A continuación se enumera la secuencia de órdenes que se puede emplear para verificar el comportamiento de la implementación de la función `link`:

```
carlos@posets:~/mount-point$ echo "Contenido del fichero" >> newfile
carlos@posets:~/mount-point$ stat newfile
  File: 'newfile'
  Size: 22          Blocks: 0          IO Block: 4096   regular file
Device: 37h/55d Inode: 2              Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/  carlos)   Gid: ( 1000/  carlos)
Access: 1970-01-01 01:00:00.000000000 +0100
Modify: 2017-03-24 23:08:56.000000000 +0100
Change: 2017-03-24 23:08:56.000000000 +0100
 Birth: -
carlos@posets:~/mount-point$ ln newfile linkfile
carlos@posets:~/mount-point$ stat newfile
  File: 'linkfile'
  Size: 22          Blocks: 0          IO Block: 4096   regular file
Device: 37h/55d Inode: 2              Links: 2
Access: (0644/-rw-r--r--)  Uid: ( 1000/  carlos)   Gid: ( 1000/  carlos)
Access: 1970-01-01 01:00:00.000000000 +0100
Modify: 2017-03-24 23:08:56.000000000 +0100
Change: 2017-03-24 23:08:56.000000000 +0100
 Birth: -
```

¹http://libfuse.github.io/doxygen/structfuse__operations.html#a1b234c43e826c6a690d80ea895a17f61

1. Una vez creado el fichero `newfile`, se puede verificar con el comando `stat` el número de enlaces a dicho fichero (**Links: 1**).
2. Una vez creado un enlace rígido con el comando `ln newfile linkfile` se observa que el número de enlaces tanto del fichero `link` como `newfile` se ha incrementado (**Links: 2**).
3. Si se borrara el fichero `newfile` o `linkfile` se eliminaría únicamente la entrada en el directorio, pero no su contenido (**Links: 1**). El contenido se eliminaría cuando se borrara el fichero con un único enlace (**Links: 1**).

2.2. Desarrollo de la parte opcional

A continuación se describirán los pasos a realizar para llevar a cabo con éxito codificación de la funcionalidad `ln` para `fuse`:

1. **[2 puntos]** Dar de alta la funcionalidad `link` para habilitar la creación de enlaces rígidos. Nota: El prototipo de `link` (`int (* fuse_operations::link)(const char* from, const char* to)`) tiene como parámetros de entrada los nombres de los ficheros involucrados en la operación `ln`, donde `from` y `to` corresponderían a `newfile` y `linkfile` del ejemplo mostrado en la sección 2.1.
2. **[4 puntos]** Desarrollo de la función `link`. Para ello se recomienda seguir los siguientes pasos:
 - Chequeo de la existencia y la longitud de los nombres de los ficheros `from` y `to`.
 - Obtención del identificador del inodo del fichero `from`, **para poder incrementar el número de enlaces de dicho inodo**. Se recomienda modificar la estructura `NodeStructure` del fichero `myFS.h` añadiendo el campo `int nlinks`; para este propósito.

```
typedef struct NodeStructure {
    int numBlocks;                // Num blocks
    int fileSize;                 // File size
    time_t modificationTime;      // Modification time
    DISK_LBA blocks[MAX_BLOCKS_PER_FILE]; // Blocks
    BOOLEAN freeNode;             // If the node is available
    int nlinks;
} NodeStruct;
```

Con el fin de ser coherentes a la hora de examinar el número de enlaces con el comando `stat` también se aconseja la modificación de la función `my_getattr` del fichero `fuseLib.c`, sustituyendo la línea `stbuf->st_nlink = 1`; por `stbuf->st_nlink = node->nlinks`;

- Creación de una nueva entrada en el directorio con el nombre del fichero `to` que contenga el mismo identificador de inodo que el fichero `from`.
3. **[1 puntos]** Modificación de la función `mknod` para que tenga en cuenta el número de enlaces de un fichero. La función `mknod` empleada cuando se crea de un fichero vacío debe de añadir el campo `nlinks=1` en el inodo asociado al fichero.
 4. **[3 puntos]** Modificación de la función `unlink` para que tenga en cuenta el número de enlaces de un fichero. La función `unlink` empleada para el borrado de un fichero deberá chequear el número de enlaces para proceder al borrado adecuado:

- Si el número de enlaces es igual a 1 entonces borrará el fichero. Es decir, eliminará los bloques asociados a dicho fichero y eliminará la entrada del nombre del fichero del directorio.
- Sin embargo, si el número de enlaces es superior a 1 únicamente eliminará la entrada del nombre del fichero en el directorio.