

UNIDAD 5

ACTIVIDAD PRÁCTICA

FICHEROS

ÍNDICE

ORGANIZACIÓN DE LA ACTIVIDAD PRÁCTICA.....	3
DESCRIPCIÓN DE LA ACTIVIDAD PRÁCTICA	3
RECOMENDACIONES E INDICACIONES PARA LA ENTREGA.....	3
RÚBRICA DE CORRECCIÓN	3
COMO REALIZAR MI ENTREGA	10

ORGANIZACIÓN DE LA ACTIVIDAD PRÁCTICA

Nombre de la práctica	
Tipo de tarea	Por equipos de 2 personas o individual
Entregables	<ul style="list-style-type: none"> • Fichero <code>simul_ext.c</code> • Link a GitHub donde pueda acceder para comprobarlo (público) • Link a un video (alojado en one drive, youtube, Dropbox, etc) de no más de 10 minutos en el cual hablen ambos integrantes del equipo, y expliquen la ejecución y el código del mismo • Indicar en la entrega si se han hecho o no cada uno de los puntos de evaluación, y si se ha tenido alguna dificultad cómo se ha resuelto.

DESCRIPCIÓN DE LA ACTIVIDAD PRÁCTICA

Esta práctica se puede hacer de manera individual o por equipos.

En esta práctica se va a desarrollar un simulador de un sistema de ficheros tipo Linux `ext` muy simplificado. En lugar de trabajar sobre una partición del disco, utilizaremos un fichero binario llamado `particion.bin` que contiene el sistema completo.

Los bloques de este sistema concreto tienen un tamaño de 512 bytes, la partición ocupa 100 bloques, y su esquema es el que se describe a continuación. No hay bloque de arranque, como en un sistema `ext2`, en el que el bloque 0 se reserva para tal fin.

El bloque 0 de `particion.bin` lo ocupa el superbloque. En el fichero `cabecera.h` que se proporciona con el enunciado, se encuentra su definición.

```
/* Estructura del superbloque */
typedef struct {
    unsigned int s_inodos_count;           /* inodos de la partición */
    unsigned int s_blocks_count;          /* bloques de la partición */
    unsigned int s_free_blocks_count;     /* bloques libres */
    unsigned int s_free_inodos_count;     /* inodos libres */
    unsigned int s_first_data_block;      /* primer bloque de datos */
    unsigned int s_block_size;            /* tamaño del bloque en bytes */
    unsigned char s_relleno[SIZE_BLOQUE-6*sizeof(unsigned int)]; /* relleno a 0's */
} EXT_SIMPLE_SUPERBLOCK;
```

Para simplificar el problema, en lugar de bitmaps, vamos a usar bytemaps o arrays de bytes para bloques e inodos. En el bytemap inodos, si el byte `M` vale 1, significa que ese inodo está ocupado, si vale 0 está libre. Lo mismo para el bytemap de bloques.

```
/* Bytemaps, caben en un bloque */
typedef struct {
    unsigned char bmap_bloques[MAX_BLOQUES_PARTICION];
    unsigned char bmap_inodos[MAX_INODOS]; /* inodos 0 y 1 reservados, inodo 2 directorio */
    unsigned char bmap_relleno[SIZE_BLOQUE - (MAX_BLOQUES_PARTICION + MAX_INODOS) * sizeof(char)];
} EXT_BYTE_MAPS;
```

El bytemap de bloques tiene 100 elementos. Los 4 primeros están ocupados por el superbloque, los bytemaps, la lista de inodos y el directorio. Los 96 restantes son los bloques de datos. El bytemap de inodos tiene 24 elementos. Los inodos 0 y 1 están reservados aunque no se usan, y el 2 corresponde al directorio.

En el bloque 3 está la lista de inodos. El inodo simplificado tiene la siguiente estructura:

```
/* inodo */
typedef struct {
    unsigned int size_fichero;
    unsigned short int i_nbloque[MAX_NUMS_BLOQUE_INODO];
} EXT_SIMPLE_INODE;
```

El inodo guarda el tamaño en bytes de información útil del fichero, si no está ocupado el valor de ese campo es 0. Además hay una lista de hasta 7 números de bloques de datos en los que se encuentra dicha carga útil. Si un fichero ocupa los bloques 5, 11 y 7, los elementos 0, 1 y 2 del array `i_nbloque` toman esos valores y el resto valen FFFFH. Para recuperar el contenido hay que concatenar los bloques 5, 11 y 7 en ese orden. La lista de inodos tiene 24 elementos.

En el bloque 3 está el directorio. Este sistema puede guardar hasta 20 ficheros. Para cada fichero hay una entrada en la que figuran el nombre del fichero y el número de inodo. La entrada 0 del directorio, contiene el nombre '.' y el número de inodo 2. El resto están vacías, y para ellas el valor de inodo es FFFFH.

```
typedef struct {
    char dir_nfich[LEN_NFICH];
    unsigned short int dir_inodo;
} EXT_ENTRADA_DIR;
```

Se pide desarrollar un programa en C, llamado `simul_ext.c`, que lea el contenido del fichero binario `particion.bin` en un array de bytes en memoria y entre en un bucle infinito de procesamiento de comandos. El programa mostrará un prompt de la forma `>>` al usuario y será capaz de manejar los comandos que se especifican a continuación (en orden de menos a más complicado).

info

Este comando muestra la información del superbloque por pantalla.

```
>> info
Bloque 512 Bytes
inodos particion = 24
inodos libres = 18
Bloques particion = 100
Bloques libres = 91
Primer bloque de datos = 4
>>
```

bytemaps

Muestra el contenido del bytemap de inodos y los 25 primeros elementos del bytemap de bloques. Este comando te resultará muy útil durante el desarrollo.

```
>> bytemaps
Inodos :1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Bloques [0-25] :1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>>
```

dir

Lista todos los ficheros (excepto la entrada especial del directorio raíz). Para cada uno mostrará su nombre, tamaño, inodo y bloques que ocupa. **Los nombres son case sensitive.**

```
>> dir
BelloGal.txt      tamaño:1289      inodo:3 bloques: 6 9 8
CHISTE.txt        tamaño:44        inodo:7 bloques: 7
OLAKEASE.txt      tamaño:11        inodo:4 bloques: 5
>>
```

rename

Cambia el nombre de un fichero en la entrada correspondiente, aunque no modifica inodos ni listas. El comando debe comprobar que el fichero origen existe y que el nombre nuevo no existe, de lo contrario dará un error.

```
>> dir
BelloGal.txt      tamaño:1289      inodo:3 bloques: 6 9 8
CHISTE.txt        tamaño:44        inodo:7 bloques: 7
OLAKEASE.txt      tamaño:11        inodo:4 bloques: 5
>> rename HOLA.txt ADIOS.txt
ERROR: Fichero HOLA.txt no encontrado
>> rename OLAKEASE.txt CHISTE.txt
ERROR: El fichero CHISTE.txt ya existe
>> rename OLAKEASE.txt HOLA.txt
>> dir
BelloGal.txt      tamaño:1289      inodo:3 bloques: 6 9 8
CHISTE.txt        tamaño:44        inodo:7 bloques: 7
HOLA.txt          tamaño:11        inodo:4 bloques: 5
>>
```

imprimir

Muestra el contenido del fichero especificado como un texto. Deberá comprobar que el fichero existe. Si el fichero ocupa más de un bloque, debe aparecer en el orden correcto.

```
>> imprimir PEPE.txt
ERROR: Fichero PEPE.txt no encontrado
>> imprimir HOLA.txt
OLA KE ASE?
>> imprimir BelloGal.txt
Gallia est omnis divisa in partes tres, quarum unam incolunt Belgae,
aliam Aquitani, tertiam qui ipsorum lingua Celtae, nostra Galli appellantur.
Hi omnes lingua, institutis, legibus inter se differunt.
Gallos ab Aquitanis Garumna flumen, a Belgis Matrona et Sequana dividit.
Horum omnium fortissimi sunt Belgae, propterea quod a cultu atque humanitate
provinciae longissime absunt, minimeque ad eos mercatores saepe commeant
atque ea quae ad effeminandos animos pertinent important, proximique sunt
Germanis, qui trans Rhenum incolunt, quibuscum continenter bellum gerunt.
Qua de causa Helvetii quoque reliquos Gallos virtute praecedunt,
quod fere cotidianis proeliis cum Germanis contendunt, cum aut suis
finibus eos prohibent aut ipsi in eorum finibus bellum gerunt.
Eorum una pars, quam Gallos obtinere dictum est, initium capit a flumine
Rhodano, continetur Garumna flumine, Oceano, finibus Belgarum, attingit
etiam ab Sequanis et Helvetiis flumen Rhenum, vergit ad septentriones.
Belgae ab extremis Galliae finibus oriuntur, pertinent ad inferiorem
partem fluminis Rheni, spectant in septentrionem et orientem solem.
Aquitania a Garumna flumine ad Pyrenaeos montes et eam partem Oceani
quae est ad Hispaniam pertinet; spectat inter occasum solis et
septentriones.
```

remove

Elimina el fichero. El programa debe marcar el inodo y los bloques libres en los bytemaps, poner tamaño 0 en el inodo liberado, marcar los 7 punteros a bloque de ese inodo con el valor FFFFH y eliminar la entrada del directorio, poniendo un string vacío en el nombre y el valor FFFFH en el número de inodo. No es necesario borrar el contenido físico de los bloques de información.

```
>> dir
BelloGal.txt    tamaño:1289    inodo:3 bloques: 6 9 8
CHISTE.txt      tamaño:44      inodo:7 bloques: 7
HOLA.txt        tamaño:11      inodo:4 bloques: 5
>> bytemaps
Inodos :1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Bloques [0-25] :1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
>> remove ADIOS.txt
ERROR: Fichero ADIOS.txt no encontrado
>> remove HOLA.txt
>> dir
BelloGal.txt    tamaño:1289    inodo:3 bloques: 6 9 8
CHISTE.txt      tamaño:44      inodo:7 bloques: 7
>> bytemaps
Inodos :1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Bloques [0-25] :1 1 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
>> █
```

copy

Copiar un fichero. Deberá comprobar que el fichero origen existe y el destino también. Para simplificar, no habrá que comprobar que se supera el número máximo de ficheros de la partición. Hay que buscar el primer inodo libre, asignárselo al fichero nuevo, copiar el tamaño y marcar el inodo como ocupado. Luego se hará un bucle sobre la lista de números de bloque del archivo origen. Se busca el primer bloque libre en el bytemap de bloques, se asigna ese número en la lista de bloques del inodo destino y se copia el contenido de los datos del bloque al primer bloque libre, que se marca como ocupado en el bytemap de bloques.

Además, se creará una entrada en la primera vacante disponible del directorio con el nombre del fichero y su número de inodo.

```
>> dir
BelloGal.txt      tamaño:1289      inodo:3 bloques: 6 9 8
CHISTE.txt        tamaño:44        inodo:7 bloques: 7
>> Copy BelloGal.txt Cesar.txt
>> dir
BelloGal.txt      tamaño:1289      inodo:3 bloques: 6 9 8
Cesar.txt         tamaño:1289      inodo:4 bloques: 4 5 10
CHISTE.txt        tamaño:44        inodo:7 bloques: 7
>> bytemaps
Inodos :1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Bloques [0-25] :1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>> imprimir Cesar.txt
Gallia est omnis divisa in partes tres, quarum unam incolunt Belgae,
aliam Aquitani, tertiam qui ipsorum lingua Celtae, nostra Galli appellantur.
Hi omnes lingua, institutis, legibus inter se differunt.
Gallos ab Aquitanis Garumna flumen, a Belgis Matrona et Sequana dividit.
Horum omnium fortissimi sunt Belgae, propterea quod a cultu atque humanitate
provinciae longissime absunt, minimeque ad eos mercatores saepe comeant
atque ea quae ad effeminandos animos pertinent important, proximique sunt
Germanis, qui trans Rhenum incolunt, quibuscum continenter bellum gerunt.
Qua de causa Helvetii quoque reliquos Gallos virtute praecedunt,
quod fere cotidianis proeliis cum Germanis contendunt, cum aut suis
finibus eos prohibent aut ipsi in eorum finibus bellum gerunt.
Eorum una pars, quam Gallos obtinere dictum est, initium capit a flumine
Rhodano, continetur Garumna flumine, Oceano, finibus Belgarum, attingit
etiam ab Sequanis et Helvetiis flumen Rhenum, vergit ad septentriones.
Belgae ab extremis Galliae finibus oriuntur, pertinent ad inferiorem
partem fluminis Rheni, spectant in septentrionem et orientem solem.
Aquitania a Garumna flumine ad Pyrenaeos montes et eam partem Oceani
quae est ad Hispaniam pertinet; spectat inter occasum solis et
septentriones.
```

En
este

ejemplo se ha copiado la céleb Cesar.txt re introducción de “De Bello Gallico” al fichero Cesar.txt. El original ocupa los bloques 6,9 y 8 y la copia los bloques 4,5 y 10. El original está descrito en el inodo 3 y la copia en el inodo 4.

salir

Este sencillo comando termina la sesión.

Comando desconocido

Si el usuario pide un comando desconocido, deberá ver este mensaje de error.

Ayuda

El alumno dispone del fichero `cabeceras.h` y de un esqueleto de lo que debe desarrollar

Entrega de la práctica

Cada equipo desarrollará el proyecto en `github`. Tiene que haber commits en `github` de los dos integrantes. El equipo subirá un fichero `README.txt` con los nombres de los estudiantes y el fuente `simul_ext.c`. Adicionalmente hay que hacer un video conjunto (podéis hacer una video llamada por `teams` y grabaros).

Elementos a entregar:

- Fichero `simul_ext.c`
- Link a GitHub donde pueda acceder para comprobarlo (público)
- Link a un video (alojado en `one drive`, `youtube`, `Dropbox`, etc) de no más de 10 minutos en el cual hablen ambos integrantes del equipo, y expliquen la ejecución y el código del mismo
- Indicar en la entrega si se han hecho o no cada uno de los puntos de evaluación, y si se ha tenido alguna dificultad cómo se ha resuelto.

Evaluación

- El programa arranca correctamente, lee el fichero `particion.bin` e inicializa las variables con el debido control de errores. Bucle de petición y lectura de comandos (1 punto)
- Comando salir y mensaje de error si no existe el comando (0,5 puntos)
- Comando `info` (0,5 puntos).
- Comando `bytemaps` (0,5 puntos).
- Comando `dir` (0,5 puntos).
- Comando `rename` (0,5 puntos).
- Comando `remove` (1 punto si funciona para ficheros de 1 bloque, 1,5 puntos para multibloque).
- Comando `imprimir` (1 punto si funciona para ficheros de 1 bloque, 1,5 puntos para multibloque).
- Comando `copiar` (1 punto si funciona para ficheros de 1 bloque, 2,5 puntos para multibloque).
- Los commits están correctamente documentados, el código está bien documentado y estructurado. El video forma parte de la documentación. (1 punto).

RÚBRICA DE CORRECCIÓN

La rúbrica para corregir el ejercicio seguirá los criterios listados a continuación, que tienen distintos pesos respecto al total de la nota.

Criterios	Excelente	Satisfactorio	No satisfactorio	Insuficiente
El programa arranca 1 pto	De 0,75 a 1 Sin problemas	De 0,5 a 0,75 Con 1 o 2 problemas menores	De 0,25 a 0,5 con más de 2 problemas menores	De 0 a 2,5 Con problemas mayores o sin entregar
Comando salir 0,5 pto	De 0,4 a 0,5 Sin problemas	De 0,3 a 0,4 Con 1 o 2 problemas menores	De 0,2 a 0,3 con más de 2 problemas menores	De 0 a 0,2 Con problemas mayores o sin entregar
Comando info (0,5 puntos).	De 0,4 a 0,5 Sin problemas	De 0,3 a 0,4 Con 1 o 2 problemas menores	De 0,2 a 0,3 con más de 2 problemas menores	De 0 a 0,2 Con problemas mayores o sin entregar
Comando bytemaps (0,5 puntos).	De 0,4 a 0,5 Sin problemas	De 0,3 a 0,4 Con 1 o 2 problemas menores	De 0,2 a 0,3 con más de 2 problemas menores	De 0 a 0,2 Con problemas mayores o sin entregar
Comando dir (0,5 puntos).	De 0,4 a 0,5 Sin problemas	De 0,3 a 0,4 Con 1 o 2 problemas menores	De 0,2 a 0,3 con más de 2 problemas menores	De 0 a 0,2 Con problemas mayores o sin entregar
Comando rename (0,5 puntos).	De 0,4 a 0,5 Sin problemas	De 0,3 a 0,4 Con 1 o 2 problemas menores	De 0,2 a 0,3 con más de 2 problemas menores	De 0 a 0,2 Con problemas mayores o sin entregar
Comando remove (1 punto si funciona para ficheros de 1 bloque, 1,5 puntos para multibloque).	De 1 a 1,5 Sin problemas y funciona para ficheros con 1 bloque o para multibloque	De 0,75 a 1 Sin problemas pero solo funciona para ficheros con 1 bloque. También si hay problemas menores en multibloque	De 0,5 a 1 con más de 2 problemas menores en 1 bloque	De 0 a 0,5 Con problemas mayores o sin entregar
Comando imprimir (1 punto si funciona para ficheros de 1 bloque, 1,5	De 1 a 1,5 Sin problemas y funciona para ficheros con 1	De 0,75 a 1 Sin problemas pero solo funciona para ficheros con 1 bloque.	De 0,5 a 1 con más de 2 problemas menores en 1 bloque	De 0 a 0,5 Con problemas mayores o sin entregar

puntos para multibloque).	bloque o para multibloque	También si hay problemas menores en multibloque		
Comando copiar (1 punto si funciona para ficheros de 1 bloque, 2,5 puntos para multibloque).	De 2 a 2,5 Sin problemas y funciona para ficheros con 1 bloque o para multibloque	De 1 a 2 correcto para un bloque y problemas menores para multibloque	De 0,5 a 1 con más de 2 problemas menores en 1 bloque	De 0 a 0,5 Con problemas mayores o sin entregar
Los commits están correctamente documentados, el código está bien documentado y estructurado. El video forma parte de la documentación	De 0,75 a 1 Sin problemas	De 0,5 a 0,75 Con 1 o 2 problemas menores	De 0,25 a 0,5 con más de 2 problemas menores	De 0 a 2,5 Con problemas mayores o sin entregar

COMO REALIZAR LA ENTREGA

- Fichero `simul_ext.c`
- Link a GitHub donde pueda acceder para comprobarlo (público)
- Link a un video (alojado en one drive, youtube, Dropbox, etc) de no más de 10 minutos en el cual hablen ambos integrantes del equipo, y expliquen la ejecución y el código del mismo
- Indicar en la entrega si se han hecho o no cada uno de los puntos de evaluación, y si se ha tenido alguna dificultad cómo se ha resuelto.