Clase Práctica - File System

Ana Felisatti

2^{do} Cuat. - 2015

Ejercicio 1

Se tiene un disco con capacidad de 128GB, con bloques de 8KB. Suponga un sistema de archivos similar a FAT, donde la tabla se ubica desde la posición 0.

Ejercicio 1

Se tiene un disco con capacidad de 128GB, con bloques de 8KB. Suponga un sistema de archivos similar a FAT, donde la tabla se ubica desde la posición 0.

(1) ¿Cuál es el tamaño que ocupará la tabla?

Ejercicio 1

Se tiene un disco con capacidad de 128GB, con bloques de 8KB. Suponga un sistema de archivos similar a FAT, donde la tabla se ubica desde la posición 0.

(1) ¿Cuál es el tamaño que ocupará la tabla?

(2) Se sabe que un archivo comienza en el bloque 20. Dada la siguiente FAT, indicar el tamaño de dicho archivo.

Bloque	0	1	2	3	4	5	6	...	20	21	22	...
Siguiente	EOF	2	23	4	5	0	7	...	21	22	3	...

Solución

(1) Vamos a tener $\frac{|disco|}{|bloque|} = \frac{2^{27}KB}{2^3KB} = 2^{24}$ entradas en la tabla (es decir, bloques totales). Entonces para identificar cada una necesitaremos 24 bits, o 3B de espacio. Luego nuestra tabla ocupará $2^{24} \times 3B = 48MB$.

(2) Siguiendo a partir de cada bloque y su sucesor vemos que el archivo ocupa los bloques 20, 21, 22, 3, 4, 5, y 0, es decir, 7 bloques. Entonces, como cada uno es de 8 KB nuestro archivo ocupa 56KB en disco. El tamaño 'real' está en el intervalo $(48, 56]$ KB pues el bloque 0 no necesariamente está lleno.

Extra

¿Cómo sabemos en verdad que el archivo empieza en el bloque 20?

Extra

¿Cómo sabemos en verdad que el archivo empieza en el bloque 20?

Gracias a los directorios.

Extra

¿Cómo sabemos en verdad que el archivo empieza en el bloque 20?

Gracias a los directorios.

Un directorio es un archivo con una entrada por cada archivo y subdirectorio que contiene. Dichas entradas tienen la metadata del archivo, como el nombre, tamaño, último acceso, etc. En particular, entre esa información se encuentra el índice de su primer bloque.

Extra

¿Cómo sabemos en verdad que el archivo empieza en el bloque 20?

Gracias a los directorios.

Un directorio es un archivo con una entrada por cada archivo y subdirectorio que contiene. Dichas entradas tienen la metadata del archivo, como el nombre, tamaño, último acceso, etc. En particular, entre esa información se encuentra el índice de su primer bloque.

El bloque del directorio **root** es distinguido, permitiendo recorrer recursivamente a partir de él cualquier ruta.

Extra

Entonces, ¿cómo encontramos el archivo /Users/afelisatti/fs.pdf?

Extra

Entonces, ¿cómo encontramos el archivo /Users/afelisatti/fs.pdf?

- Cargamos el bloque del directorio *root* (que sabemos donde está).

Extra

Entonces, ¿cómo encontramos el archivo /Users/afelisatti/fs.pdf?

- Cargamos el bloque del directorio *root* (que sabemos donde está).
- Buscamos ahí la entrada para el directorio *Users* que nos dira dónde está su bloque inicial.

Extra

Entonces, ¿cómo encontramos el archivo /Users/afelisatti/fs.pdf?

- Cargamos el bloque del directorio *root* (que sabemos donde está).
- Buscamos ahí la entrada para el directorio *Users* que nos dira dónde está su bloque inicial.
- Cargamos entonces el bloque de *Users*.

Extra

Entonces, ¿cómo encontramos el archivo /Users/afelisatti/fs.pdf?

- Cargamos el bloque del directorio *root* (que sabemos donde está).
- Buscamos ahí la entrada para el directorio *Users* que nos dira dónde está su bloque inicial.
- Cargamos entonces el bloque de *Users*.
- Buscamos la entrada para *afelisatti* y su bloque inicial.

Extra

Entonces, ¿cómo encontramos el archivo /Users/afelisatti/fs.pdf?

- Cargamos el bloque del directorio *root* (que sabemos donde está).
- Buscamos ahí la entrada para el directorio *Users* que nos dira dónde está su bloque inicial.
- Cargamos entonces el bloque de *Users*.
- Buscamos la entrada para *afelisatti* y su bloque inicial.
- Cargamos el bloque de *afelisatti*.

Extra

Entonces, ¿cómo encontramos el archivo /Users/afelisatti/fs.pdf?

- Cargamos el bloque del directorio *root* (que sabemos donde está).
- Buscamos ahí la entrada para el directorio *Users* que nos dira dónde está su bloque inicial.
- Cargamos entonces el bloque de *Users*.
- Buscamos la entrada para *afelisatti* y su bloque inicial.
- Cargamos el bloque de *afelisatti*.
- Buscamos la entrada para *fs.pdf*.

Extra

Entonces, ¿cómo encontramos el archivo /Users/afelisatti/fs.pdf?

- Cargamos el bloque del directorio *root* (que sabemos donde está).
- Buscamos ahí la entrada para el directorio *Users* que nos dira dónde está su bloque inicial.
- Cargamos entonces el bloque de *Users*.
- Buscamos la entrada para *afelisatti* y su bloque inicial.
- Cargamos el bloque de *afelisatti*.
- Buscamos la entrada para *fs.pdf*.

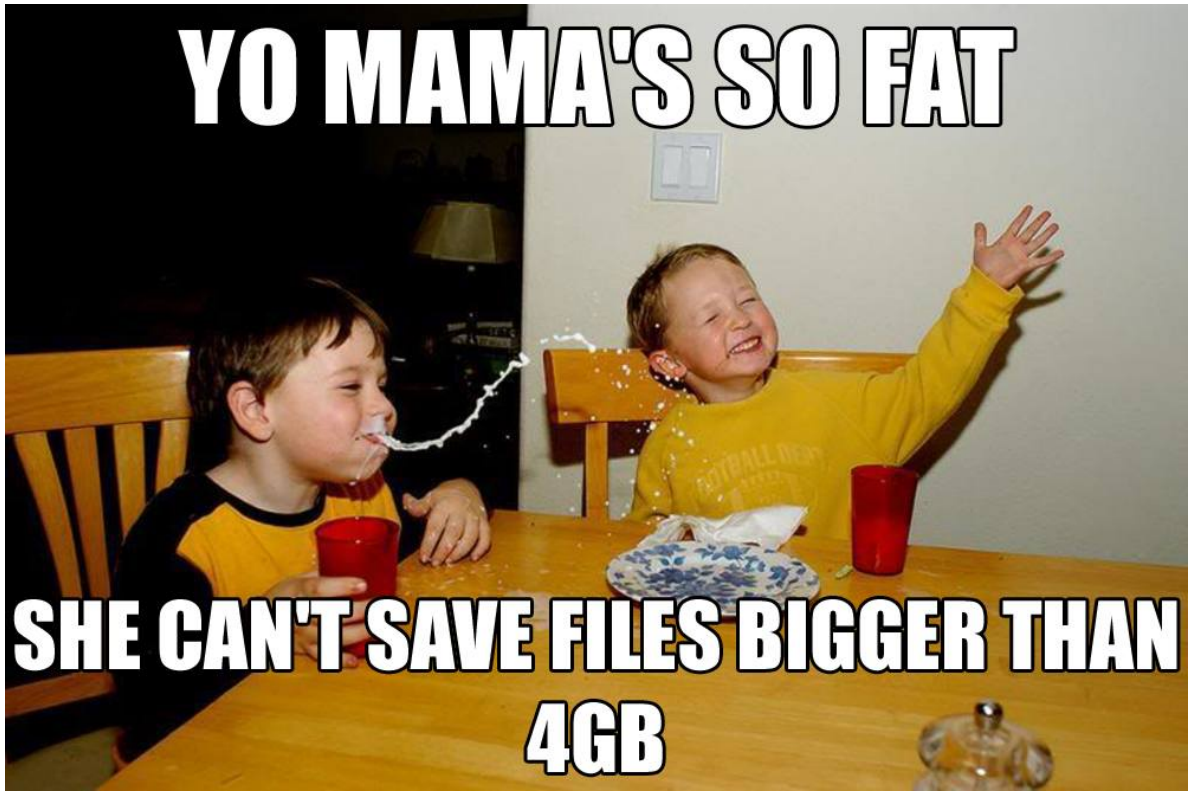
Notar que podríamos tener que cargar más de un bloque por directorio si estos fueran grandes.

Extra

¿Cuál es el máximo tamaño de archivo posible con FAT32?

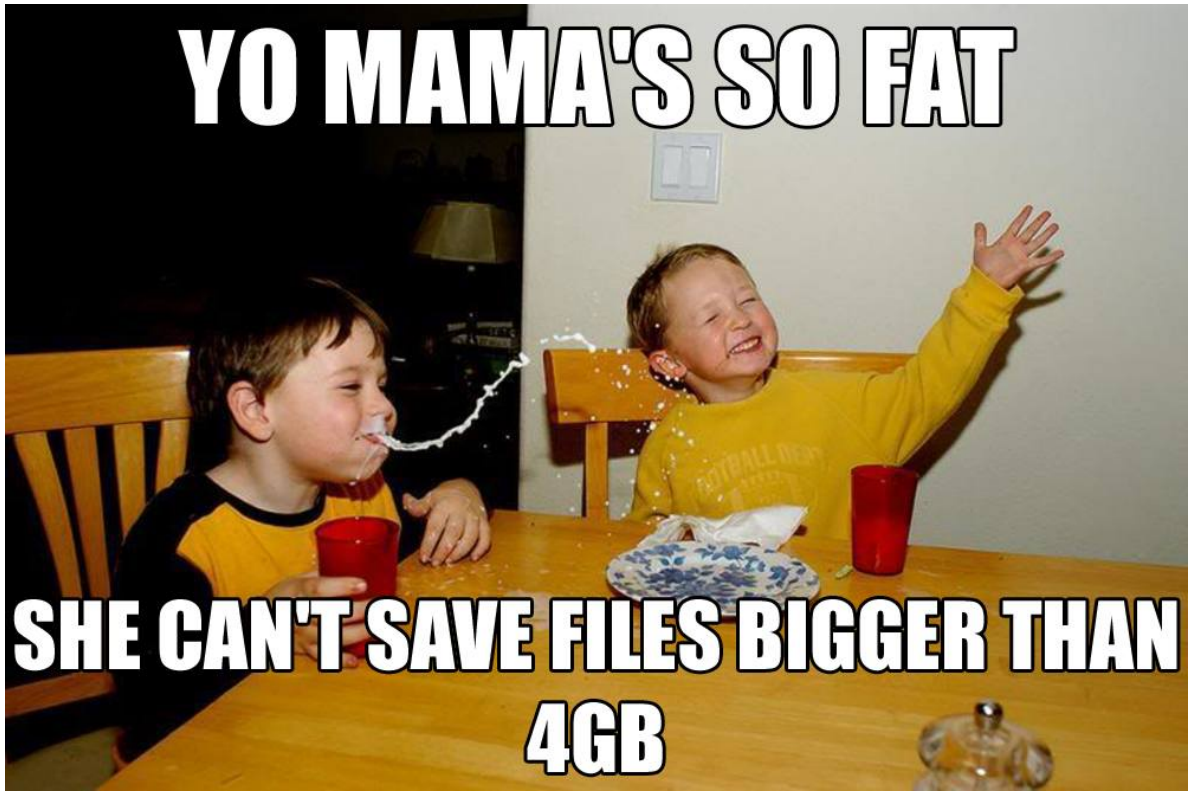
Extra

¿Cuál es el máximo tamaño de archivo posible con FAT32?



Extra

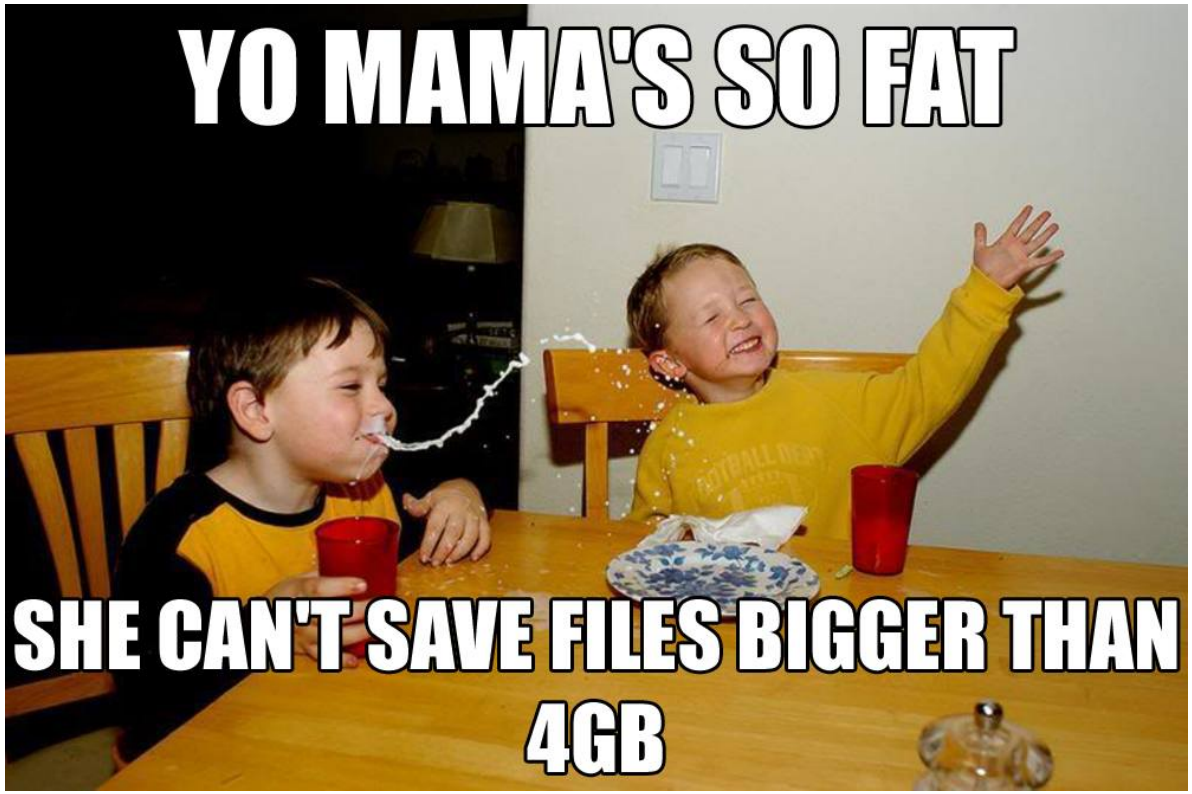
¿Cuál es el máximo tamaño de archivo posible con FAT32?



¿Por qué?

Extra

¿Cuál es el máximo tamaño de archivo posible con FAT32?



¿Por qué? Porque el campo tamaño (en bytes) en las entradas de directorio tiene 32 bits.

Ejercicio 2

Un disco posee bloques de 8KB. Un FS está administrando un archivo que ocupa 10 MB. El usuario del archivo desea insertar un bloque de 8KB con datos a la mitad del archivo. Suponiendo que el usuario ya se encuentra posicionado en la mitad del archivo y que cada bloque en el disco se direcciona con 4B ¿cuántos accesos de lectura y de escritura realiza el FS si ...

Ejercicio 2

Un disco posee bloques de 8KB. Un FS está administrando un archivo que ocupa 10 MB. El usuario del archivo desea insertar un bloque de 8KB con datos a la mitad del archivo. Suponiendo que el usuario ya se encuentra posicionado en la mitad del archivo y que cada bloque en el disco se direcciona con 4B ¿cuántos accesos de lectura y de escritura realiza el FS si ...

(1) ... los archivos se almacenan consecutivamente?

Ejercicio 2

Un disco posee bloques de 8KB. Un FS está administrando un archivo que ocupa 10 MB. El usuario del archivo desea insertar un bloque de 8KB con datos a la mitad del archivo. Suponiendo que el usuario ya se encuentra posicionado en la mitad del archivo y que cada bloque en el disco se direcciona con 4B ¿cuántos accesos de lectura y de escritura realiza el FS si ...

(1) ... los archivos se almacenan consecutivamente?

(2) ... si el FS usa una tabla de alocación ?

Ejercicio 2

Un disco posee bloques de 8KB. Un FS está administrando un archivo que ocupa 10 MB. El usuario del archivo desea insertar un bloque de 8KB con datos a la mitad del archivo. Suponiendo que el usuario ya se encuentra posicionado en la mitad del archivo y que cada bloque en el disco se direcciona con 4B ¿cuántos accesos de lectura y de escritura realiza el FS si ...

(1) ... los archivos se almacenan consecutivamente?

(2) ... si el FS usa una tabla de alocación ?

(3) ... si el FS usa inodes ?

Solución

El archivo ocupa $\frac{|archivo|}{|bloque|} = \frac{5 \times 2^{11} KB}{2^3 KB} = 5 \times 2^8 = 1280$ bloques y queremos meter un bloque entero en medio (ya estamos parados en el bloque 640). Como en todos los casos tendremos el costo de escribir el nuevo bloque, no lo vamos a tener en cuenta.

(1) Si el archivo se almacena consecutivamente, esto quiere decir que se deben transladar los 640 bloques para hacer lugar. Esto es: cada bloque tiene que ser leído y escrito en el bloque siguiente. Osea, tenemos que leer y escribir $640 \times 8KB$ datos. Como ya dijimos estos son 640 bloques que leemos y escribimos.

Solución

(2) Teniendo una tabla de alocación simplemente debemos actualizar las referencias al bloque siguiente para el de la mitad. Esto es: leer la entrada para ver quién era el bloque $mitad + 1$, colocarlo como sucesor del bloque nuevo y poner al nuevo como sucesor del bloque mitad. Entonces leemos 4B y escribimos 8B. Las entradas de los bloques a modificar podrían estar en bloques distintos por lo cual leeremos/escribiremos entre 1 y 2 bloques de disco.

(3) Con inodos debemos transladar las referencias a los 640 bloques en la primer tabla de indireccionamiento un lugar para poder agregar la referencia al nuevo bloque. Luego, debemos leer y escribir $640 \times 4B$ y escribir los $4B$ de la nueva referencia. Todos estos datos estan dentro del mismo bloque osea que solo leeremos y escribiremos en 1.

Extra

¿De dónde sacamos el inodo del archivo?

Extra

¿De dónde sacamos el inodo del archivo?

De los directorios nuevamente.

Extra

¿De dónde sacamos el inodo del archivo?

De los directorios nuevamente.

Como en FAT, un directorio es un archivo con una entrada por cada archivo y subdirectorio que contiene. Pero las entradas ahora solo tienen el nombre del archivo y el identificador de inodo que le corresponde (el resto de la metadata está dentro del inodo).

Extra

¿De dónde sacamos el inodo del archivo?

De los directorios nuevamente.

Como en FAT, un directorio es un archivo con una entrada por cada archivo y subdirectorio que contiene. Pero las entradas ahora solo tienen el nombre del archivo y el identificador de inodo que le corresponde (el resto de la metadata está dentro del inodo).

El inodo del directorio **root** es distinguido, permitiendo recorrer recursivamente a partir de él cualquier ruta, como en FAT.

Fin

¿Preguntas?