Manejo de Archivos

Conceptos Básicos

Particiones

Son una división lógica de un almacenamiento secundario.

Existen varios tipos y entre ellos están:

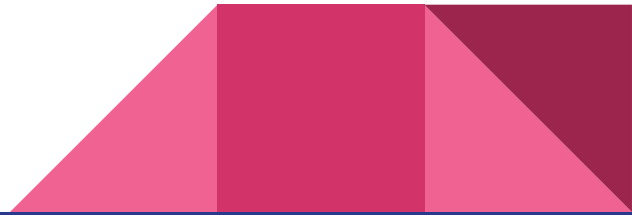
- Primarias:En la práctica se cambian los datos de la estructura del MBR.
- Extendidas: Sirven para almacenar las particiones lógicas.
- Lógicas: Son las que se escriben en una nueva estructura llamada EBR.

Limitaciones

Dentro del disco solamente pueden existir 4 particiones y existen 2 casos para estas y son:

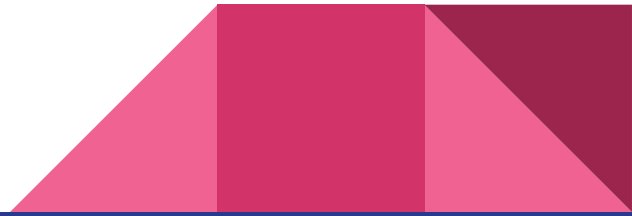
- Cuando existen 4 particiones primarias.
- Cuando existen 3 primarias y 1 extendida.

La partición extendida puede almacenar n cantidad de particiones lógicas, la única limitante que posee es el tamaño de la partición extendida.



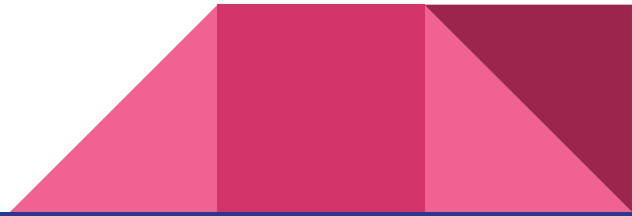
Super Bloque

Es la estructura que contiene la información más importante, describe el estado de los sistemas de los ficheros.



BitMap

Un mapa de bits, es una representación binaria en el cual un bit o conjunto de bits representan un objeto específico. En este caso se utiliza para indicar el estado de los inodos/bloques para ver si están ocupados o libres.



Sistema de Archivos (ext2/ext3)

Para la creación del sistema de archivos se utiliza la fórmula en donde se debe despejar “n”, con el objetivo de mapear y dividir la partición, para saber en qué posición se debe escribir cada cosa, toda información se guarda en el super bloque.

$$tamParticion = sizeof(superbloque) + n + 3n + n * sizeof(inodo) + 3n * sizeof(block)$$

El mapeo se conoce como formateo.

Sistema de Archivos (ext2/ext3)

Para el sistema de archivos se crea de manera inicial una carpeta root “/”.
Se comienza con utilizar la fórmula que se necesita (ext3 o ext2) para el despeje de n.

- Se escribe al principio de la partición el Super Bloque con la información necesario, como la que se sacó con el despeje de n
- Se escribe al principio de la partición el Super Bloque con la información necesaria, como la que se sacó con el despeje de n.
- Se va a la posición donde inicia el bitmap de bloques, y de inicio a fin de este se escribe '0'.
- Se crea la carpeta raíz ("/").
- Se hacen todos los cambios necesarios cuando se crea una carpeta.

Fseek

Es una función que permite desplazar el indicador de posición de fichero al sitio desde donde el cual quieres acceder al fichero.

Sintaxis

fseek(FILE *stream, long offset, int whence);

stream:Puntero del archivo.

offset :Número de bytes o caracteres que serán movidos desde la posición actual del archivo.

whence:Puntero de la posición actual del archivo desde donde el offset será añadido. Este tiene tres constantes a especificar.

Whence

Este parámetro se divide en tres constantes las cuales son:

SEEK_SET: Mueve la posición del puntero al principio del archivo.

SEEK_CUR: Mueve la posición del puntero a la posición específica.

SEEK_END: Mueve la posición del puntero al final del archivo.

fwrite

Escribe una matriz de elementos de recuento , cada uno con un tamaño de bytes de tamaño , desde el bloque de memoria apuntado por ptr hasta la posición actual en la secuencia .

Sintaxis

fwrite (const void * ptr, size_t size, size_t count, FILE * stream);

ptr:Puntero a la matriz de elementos que se van a escribir, convertida en const void*

size_t size: Tamaño en bytes de cada elemento a escribir.

size_t count: Número de elementos, cada uno con un tamaño de bytes de tamaño .

stream:Puntero del archivo.

fwrite

Escribe una matriz de elementos de recuento , cada uno con un tamaño de bytes de tamaño , desde el bloque de memoria apuntado por ptr hasta la posición actual en la secuencia .

Sintaxis

fwrite (const void * ptr, size_t size, size_t count, FILE * stream);

ptr:Puntero a la matriz de elementos que se van a escribir, convertida en const void*

size_t size: Tamaño en bytes de cada elemento a escribir.

size_t count: Número de elementos, cada uno con un tamaño de bytes de tamaño .

stream:Puntero del archivo.