file systems

disk = linear sequence of fixed-size blocks can be read and written.

what must a file system do? → implements file, directory, permission, meta data (file name, size..) ...

different filesystem design issues → file to block mapping, att. representation, dir organization, free block management

fragmentation = unused areas on disk

↳ internal = unused areas inside a block → happens on all file systems

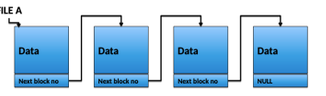
↳ external = unused areas between blocks → if new big size file needs to added, you have to shift others

file block management

↳ contiguous allocation = easy to implement and fast sequential access (used in DVD's/CD's write once read mult)

- no external fragmentation, but if you want to add to a file, need to update others too

↳ linked list allocation = can update one easily, but need to access all to reach only the last one (can store the file in different blocks independent of the external fragment)



↳ file allocation table = linked list implemented on an array

- each entry consist a pointer to block (not the block itself) → no need to access/trace whole blocks to reach one keeps a separate free list for free blocks

- for huge disk sizes the table size is huge → block size can be incremented but then, internal fragmentation

↳ indexed mapping = used in Linux, keep a tree of block pointers in i-node (index node) per file

- each file has i-node, inode has direct pointers, indirect, double, triple pointers to blocks that consist the data

block size = you can cluster blocks to make their size bigger (cannot make it smaller than its actual size)

free block management =

↳ bitmap = 0 for free, 1 for in use for all blocks, may need to scan whole array to find a empty block

- bitmap (fixed size) size is also huge, cached to RAM, if shutted down before rewrite → integrity issues, lost information

↳ free list = linked list of free blocks addresses (x 4Bytes larger, but get smaller as disk is used)

file attributes = name, identifier, type, location, size, permissions, timestrap, credentials...

- FAT = keeps att in directory structure, and starting block id, next can be accessed through tracing (4 GB limit)
- LINUX = name + i-node index per file, i-node includes all attributes and blocks

links = when the same file is under two different directories

↳ hard link = LINUX, they point to same file position (same inode index)

↳ soft link = WINDOWS, redirection (when original one is deleted, linked one disappears)