

Local File Systems

Exercise 1: FAT-Based Approach

- Given: A (very small) hard disk with blocks 0 - 11.
 - File A.TXT is stored in disk blocks 3, 1, 7 (in this order).
 - File B.DOC is stored in disk blocks 2, 11, 0, 10 (in this order).
 - Disk block 5 is damaged.
 - The remaining blocks are free.
 - Block length: 1 KByte (= 1024 Byte).
- Do the following:
 - Draw a directory that contains A.TXT and B.DOC.
 - Show for each file only the file name, the number of the first physical disk block, and a possible length in bytes.
 - Draw the corresponding FAT.
 - Assume that byte no. 3525 of file B.DOC shall be accessed:
 - How does the search for this byte proceed?
 - Which disk block and offset (= byte number within the block) result from the search?
 - What happens
 - when A.TXT is shortened by one block?
 - when B.DOC is extended by one block?
 - when B.DOC is deleted?

p.t.o.

Exercise 2: Inode-Based Approach

- Given: Two files A.TXT and B.DOC
 - File A.TXT is stored in disk blocks 3, 1, 7 (in this order).
 - File B.DOC is stored in disk blocks 2, 4, 5, 0 (in this order).
 - Block length: 1 KByte (= 1024 Byte).
- Do the following (using the traditional UNIX implementation):
 - Draw a directory containing A.TXT and B.DOC.
 - Draw the corresponding inodes.
 - Draw only those portions that specify a possible length in bytes and all the numbers of the physical disk blocks
 - Let m (the number of direct block addresses in the inode) be 10
 - Assume that byte no. 2100 of file A.TXT shall be accessed:
 - How does the search for this byte proceed?
 - Which disk block and offset (= byte number within the block) result from the search?
 - What happens
 - when A.TXT is shortened by one block?
 - when B.DOC is extended by
 - one block?
 - and eight more blocks?
 - (and how would this be done with the modern implementation based on "extents"?)
 - when B.DOC is deleted?