

Local File Systems – Solution

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.

Solution: (the block length is one of multiple possible values)

A	TXT	...	3	3000	B	DOC	...	2	4000
---	-----	-----	---	------	---	-----	-----	---	------

- Draw the corresponding FAT.

Solution:

0	1	2	3	4	5	6	7	8	9	10	11
10	7	11	1	free	bad	free	eof	free	free	eof	0

- 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?

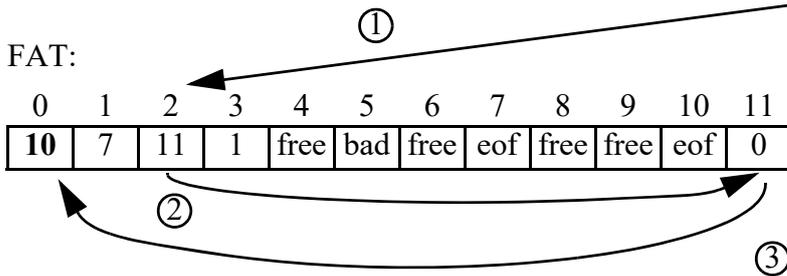
Solution:

Access logical byte 3525

→ in logical block 3 (because $3 = 3525 / 1024$, assuming block size 1K)

Directory entry:

A	TXT	...	3	3000	B	DOC	...	2	4000
---	-----	-----	---	------	---	-----	-----	---	------



→ physical block 10, offset 453 ($453 = 3525 \text{ mod } 1024$)

- What happens
 - when A.TXT is shortened by one block?

Solution: *(the block length is one of multiple possible values)*

A	TXT	...	3	2000	B	DOC	...	2	4000
---	-----	-----	---	------	---	-----	-----	---	------

0	1	2	3	4	5	6	7	8	9	10	11
10	eof	11	1	free	bad	free	free	free	free	eof	0

- when B.DOC is extended by one block?

Solution: *(the block length is one of multiple possible values)*

A	TXT	...	3	3000	B	DOC	...	2	5000
---	-----	-----	---	------	---	-----	-----	---	------

0	1	2	3	4	5	6	7	8	9	10	11
10	7	11	1	eof	bad	free	eof	free	free	4	0

- when B.DOC is deleted?

Solution:

A	TXT	...	3	3000
---	-----	-----	---	------

0	1	2	3	4	5	6	7	8	9	10	11
free	7	free	1	free	bad	free	eof	free	free	free	free

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.

Solution: (the inode numbers are just examples)

4711	A.TXT	4805	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

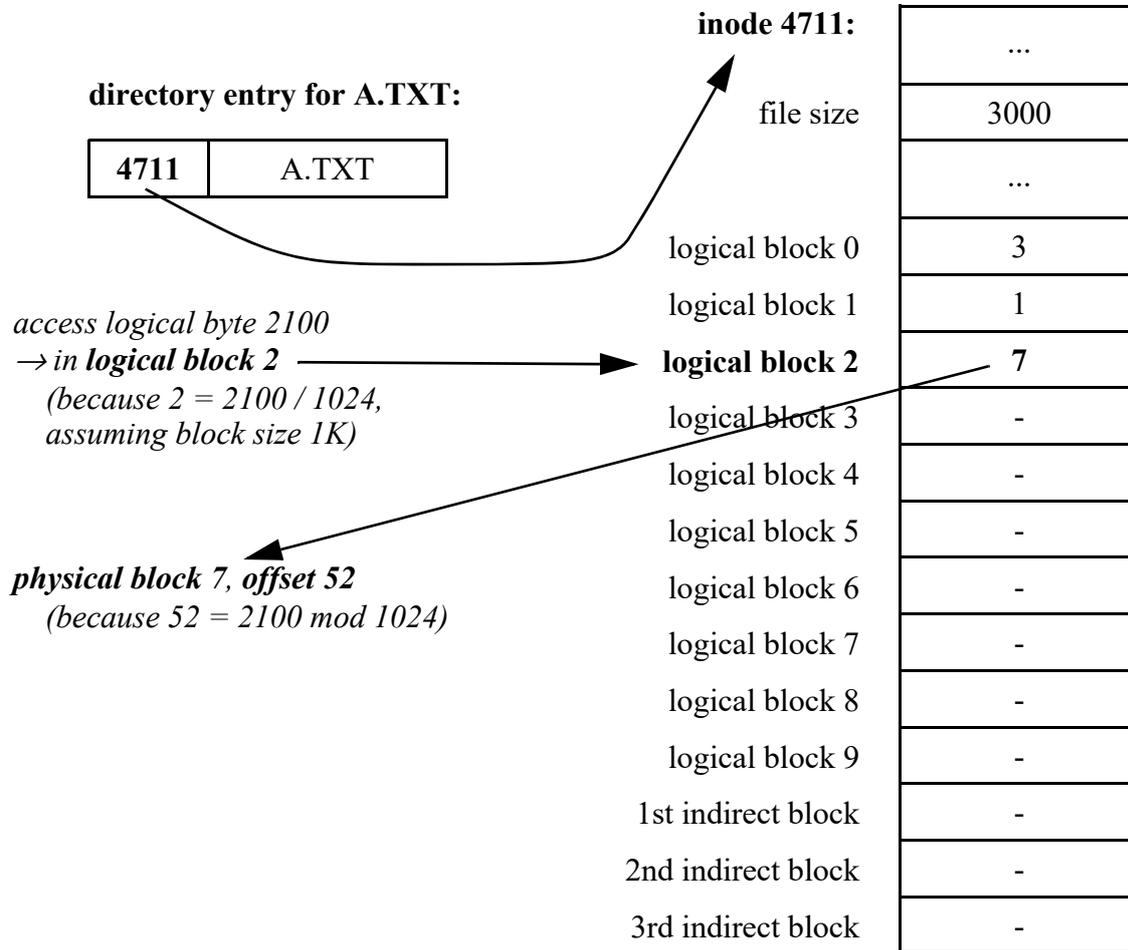
Solution: (the file lengths and block numbers are just examples for possible values)

<u>inode 4711:</u>	...	<u>inode 4805:</u>	...
file size	3000		4000

	3		2
	1		4
	7		5
	-		0
10 direct blocks	-		-
	-		-
	-		-
	-		-
	-		-
	-		-
1st indirect block	-		-
2nd indirect block	-		-
3rd indirect block	-		-

- 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?

Solution:



- What happens
 - when A.TXT is shortened by one block?
 - when B.DOC is extended by
 - one block?

Solution: (the file lengths and the new block no. are just examples for possible values)

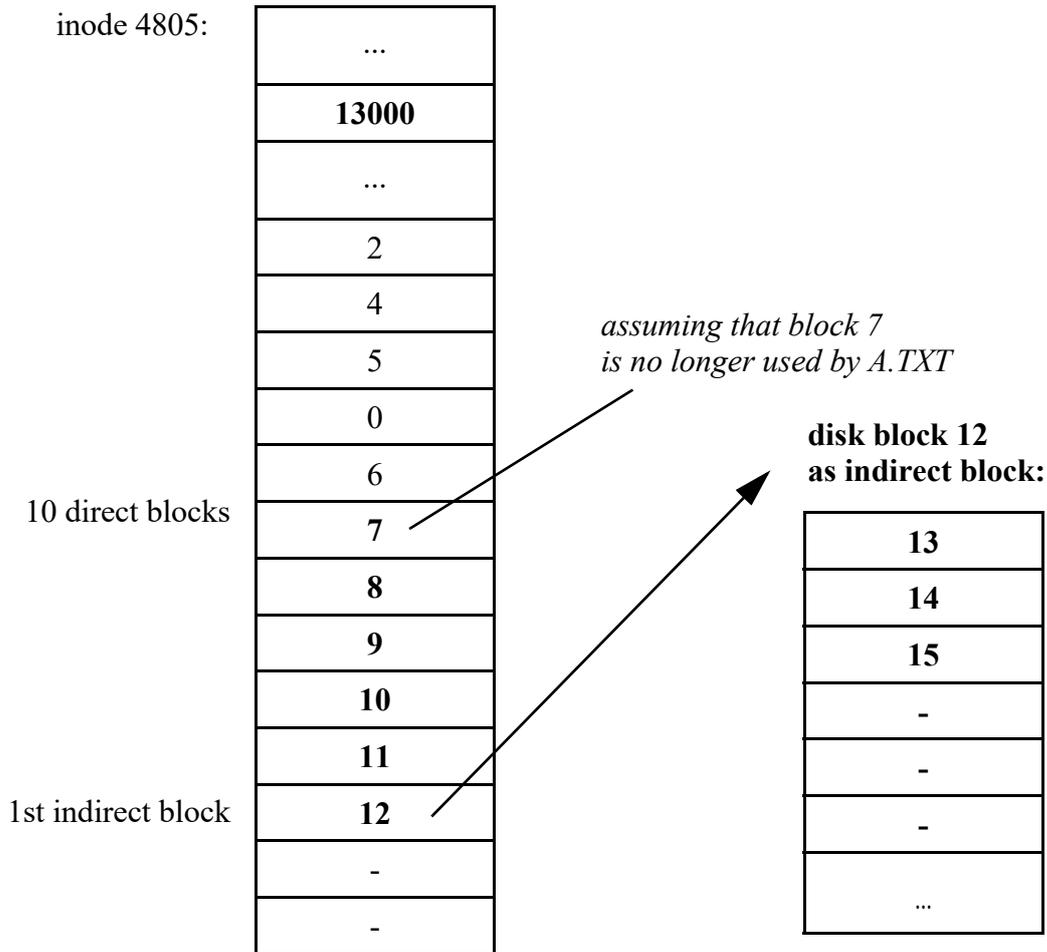
	<u>inode 4711:</u>	<u>inode 4805:</u>

file size	2000	5000

	3	2
	1	4
	-	5
	-	0
10 direct blocks	-	6
	-	-
	-	-
	-	-
	-	-
	-	-
1st indirect block	-	-
2nd indirect block	-	-
3rd indirect block	-	-

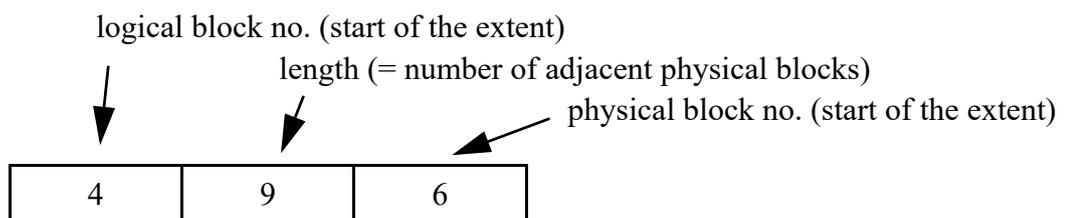
- and eight more blocks?

Solution:



- (and how would this be done with the modern implementation based on "extents"?)

Solution: *(just the extent for the last blocks as an example)*



*(i.e.: the 9 logical blocks starting from logical block no. 4
are stored in the 9 adjacent physical blocks starting from physical block 6;
no indirect block needed)*

- when B.DOC is deleted?

Solution: The directory entry is deleted. The inode is marked to be free for reuse.