

Exercise

Exercise 3 (3.5 points).

Given the file system of the figure, which has the following characteristics:

- Block size: 1024 bytes.
- Size of a block address: 2 bytes.
- Number of sectors per block: 2
- Read time for a sector: 1 ms.
- Each i-node needs a block.
- i-node field:
 - ID of an i-node (ID)
 - Metadatos (file attributes, owner and group ID, etc.)
 - Type of the element: directory (dir), file (file) or link (lnk).
 - Link counter (CE)
 - 1 direct pointer (PD),
 - 1 single indirect pointer (PIS)
 - 1 double indirect pointer (PID).

The following figure shows a system configuration file. A blank value means that the associated input is empty (is type void / null).

Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8
Superblock root i-node: 0	ID: 0 Metadata Type: dir CE: 3 PD: 51 PIS: PID:	ID: 1 Metadata Type: file CE: 1 PD: 100 PIS: PID:	ID: 2 Metadata Type: file CE: 1 PD: 103 PIS: 52 PID:	ID: 3 Metadata Type: dir CE: 4 PD: 53 PIS: PID:	ID: 4 Metadata Type: dir CE: 2 PD: 54 PIS: PID:	ID: 5 Metadata Type: lnk CE: 1 PD: 55 PIS: PID:	ID: 6 Metadata Type: dir CE: 2 PD: 56 PIS: PID:	ID: 7 Metadata Type: file CE: 1 PD: 120 PIS: 121 PID: 57

Block 51	Block 52	Block 53	Block 54	Block 55	Block 56	Block 57	Block 58	Block 59
. 0	104	. 3	. 4	/Murcia	. 6	130	131	
.. 0	105	.. 0	.. 3		.. 3	58	132	
Madrid 1	106	Norte 4	Ciudad 5				133	
Lugo 2		Sur 6						
Murcia 3		Centro 7						

State:

1. Represent the tree structure of files/directories. Which problem can exist on exploring the previous structure in the search for a file?
2. What is the maximum size of a file?
3. Describe how the following operation is performed and what changes would be made in the filesystem.

`rm /Murcia/Norte/Ciudad`



4. Calculate the time to read the first byte of the file /Madrid
5. Calcule el tiempo necesario para leer el último byte del fichero /Murcia/Centro

NOTE: The command *rm* remove a file.

SOLUTION:

1. Represent the tree structure of files/directories. Which problem can exist on exploring the previous structure in the search for a file?

/Madrid	(File)
/Lugo	(File)
/Murcia/Norte/Ciudad	(Link)
/Murcia/Sur	(Empty directory)
/Murcia/Centro	(File)

The problem is the circular reference using this link:

/Murcia/Norte/LNK/Murcia

Two possible solutions: do not use symbolic links in searches and restrict the number of directories.

2. What is the maximum size of a file?

A direct pointer addresses a block 1KB.

Since the block size is 1KB size and direction are 2B, each single indirect point addresses a block containing $1KB / 2B = 512$ direct pointers, being able to address 512KB in total.

A double indirect pointer addresses a block containing $1KB / 2B = 512$ single indirect pointers, each of which addresses 512KB. May address: $512 * 512 KB = 256 MB$

The maximum file size is: 512KB 1KB + + 256 MB..

3. Describe how the following operation is performed and what changes would be made in the filesystem.

`rm /Murcia/Norte/Ciudad`

We access the block 0 and the root inode is identified, we access the i-node 0 (block 1) and its contents (block 51). Next, we access the i-node 3 (block 4) and its content (block 53). We access to the i-node 4 (block 5) and its content (block 54). Finally,

we access to the i-node 5 (block 6). Note that the link count is 1, so the link can be deleted. Both 5 inode as block 55 remain free. The result is:

Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8
Superblock root i-node: 0	ID: 0 Metadata Type: dir CE: 3 PD: 51 PIS: PID:	ID: 1 Metadata Type: fil CE: 1 PD: 100 PIS: PID:	ID: 2 Metadata Type: file CE: 1 PD: 103 PIS: 52 PID:	ID: 3 Metadata Type: dir CE: 4 PD: 53 PIS: PID:	ID: 4 Metadata Type: dir CE: 2 PD: 54 PIS: PID:	ID: 5 Metadata Type: lnk CE: 1 PD: 55 PIS: PID:	ID: 6 Metadata Type: dir CE: 2 PD: 56 PIS: PID:	ID: 7 Metadata Type: file CE: 1 PD: 120 PIS: 121 PID: 57

Block 51	Block 52	Block 53	Block 54	Block 55	Block 56	Block 57	Block 58	Block 59
. 0	104	. 3	. 4	/Murcia	. 6	130	131	
.. 0	105	.. 0	.. 3		.. 3	58	132	
Madrid 1	106	Norte 4	Ciudad 5				133	
Lugo 2		Sur 6						
Murcia 3		Centro 7						

4. Calculate the time to read the first byte of the file /Madrid

Accessing to the file requires the following steps:

First, we access the block 0 and root inode is identified. Then, we access the i-node 0 (block 1) and its contents (block 51). Next, we access to the i-node 1 (block 2) and its content (block 100). In total, 5 blocks are accessed, representing 10 sectors, which takes 10 ms.

5. Calculate the time to read the last byte of the file /Murcia/Center

We access the block 0 and the root inode is identified. Next, we access the i-node 0 (block 1) and its contents (block 51). We access to the i-node 3 (block 4) and its content (block 53). We access to the i-node 7 (block 8). The last byte of the file is associated with the double indirect pointer. So, it is necessary to access the block 57 and through the last entry in the same block 58. The last entry of the same points to the last data block is 133.

So we need to access: 9 data blocks that represent 18 sectors associated with a time of 18 ms.