Tlemcen University

Department of Computer Science

1st Year Engineering

" Introduction to operating systems 1"

Session 6: UNIX file systems

Academic year: 2023-2024

Outline

- ❑ File systems
- ❑ i-nodes (inodes)
- ❑ Physical and symbolic links

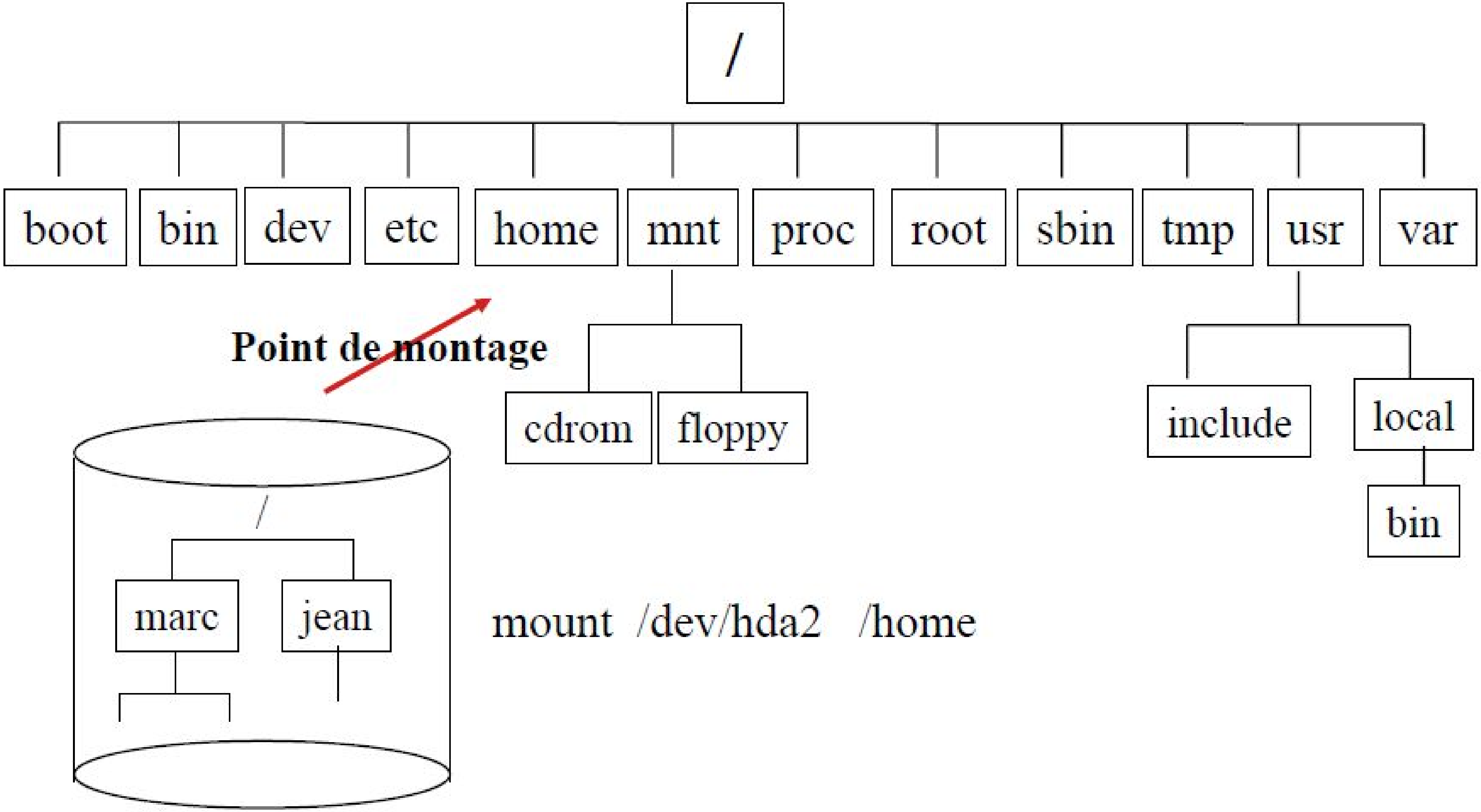
File systems / Unix

- File systems / Unix
  - *A single tree structure*
  - *Composed of one or more file systems*
  - *Possible assembly: single system*
  - *A file system accessible on disk by:*
    - *a logical device number*

      system no. = device no.
    - *physical addresses calculated by the driver*
  - *The Unix file system mounted on the / root*
    - *fstab file is a fil in which we find the sequence of file systems to be mounted in order. (this file is placed in /etc).*
    - *vfs (virtual file system): enables the Linux kernel to recognise the file system*
  - *Several file system variants*

- Examples of file systems on Linux
  - *Unlogged (no recovery log in the event of failure)*
    - *Ext2fs*
    - *FAT12*
    - *FAT16*
    - *FAT32*
  - *Logged*
    - *ext3fs*
    - *XFS*
    - *NTFS….*
  - *And for networks NFS….*

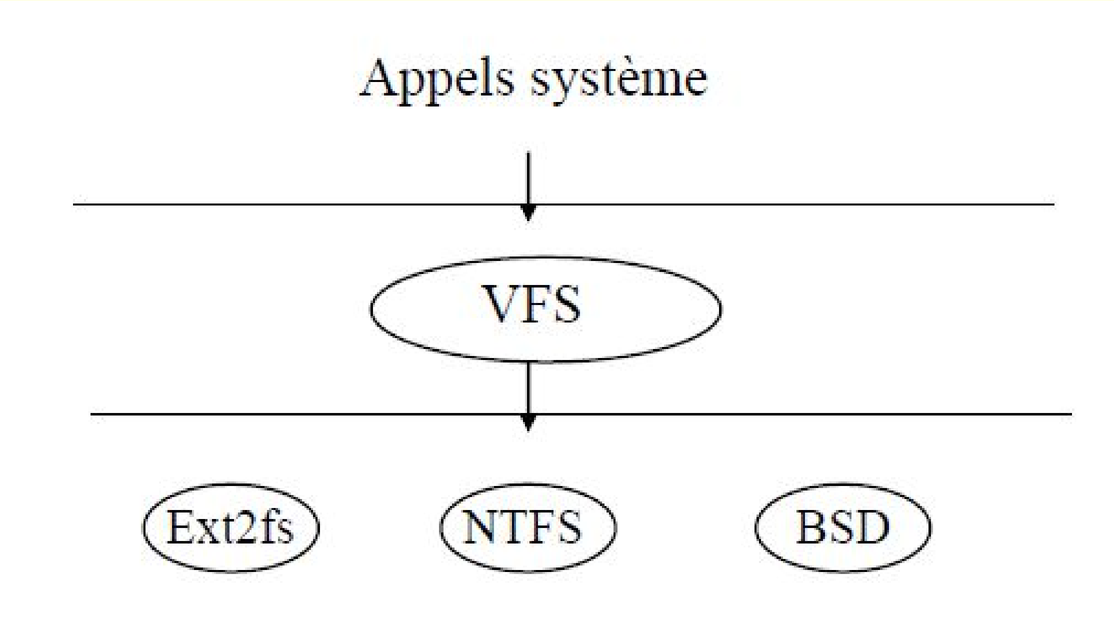
❑ The tree structure FHS (File Hierarchy Standard)



mount /dev/hda2 /home

❑ VFS (virtual file system)

- *Allows several file management systems to coexist.*
- *This is a software layer inserted into the kernel above the file systems.*



System calls: read, write, open, close files

Files

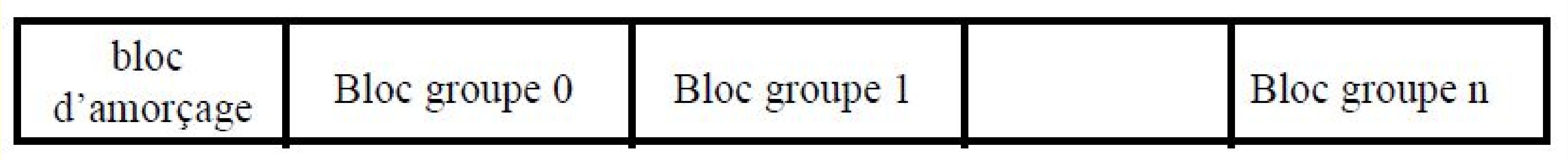
- There are three types of file:
  - ***Data files**: a set of data*
    - *directory (d)*
    - *regular or ordinary: programs, data.... (-)*
    - *symbolic links (l)*
  - ***Exchange files**: for data communication*
    - *FIFO files or pipes (p): for communication between processes*
    - *Socket (s)*
  - ***Hardware file**: peripheral resource*
    - *Block files (b): hard disk, CD-ROM, etc.*
    - *Character files (c): screen, keyboard, mouse, etc.*

Disk organization

- ***Block***
  - *exchange unit: 1024... Bytes*
- ***System blocks***
  - ***boot**: 1st block or other blocks on disk*
  - ***super-block**: file system status*
    - *no. of first free blocks*
    - *list of free i-nodes*
- ***i-node table** configured during installation*
- ***Data blocks***
- ***i-node** is a descriptor storing file information on disk*

❑Example: allocating Linux Ext2fs blocks

- *Partition into groups of blocks*
- *Choice of block sizes: 1024 to 4096*
- *A partition is composed of :*
  - *a boot block*
  - *a set of block groups with data blocks*
  - *Maximum size of a partition 4TO*



bloc d'amorçage	Bloc groupe 0	Bloc groupe 1		Bloc groupe n

- *Each group of Ext2 blocks contains:*
  - *A copy of the file system superblock (1 block)*
  - *A copy of the block group descriptors (n blocks)*
  - *Group block occupancy table (bitmap) (1 block)*
  - *Group i-node occupancy table (bitmap) (1 block)*
  - *Group i-nodes table (n blocks)*
  - *File data blocks (n blocks)*

❑ Example:

- *If a partition is 4GB, a block is 2KB.*
  - ➢ *Data block bitmap = 8(bits) * 2K = 16K bits because 1 bit/block*
  - ➢ *Number of blocks in a group: 16K data blocks (16384 blocks)*
  - ➢ *Number of bytes of data in a group: 2KB (1 block) * 16K = 32MB*
  - ➢ *For 4GB you need 4GB/32MB = 128 groups of blocks*

❑ The superblock - Managing disk space

- *The size of the file system (number of blocks)*
- *The number of free blocks in the file system*
- *A list of free blocks available in the file system*
- *The index of the 1st free block in the list of free blocks*
- *The size of the i-node table*
- *The number of free i-nodes in the file system*
- *A list of free i-nodes in the file system*
- *The index of the 1st free inode in the list of free inodes*
- *Lock fields for the lists of free blocks and inodes*
- *A flag indicating whether the superblock has been modified*

❑Example: Linux Ext2 superblock

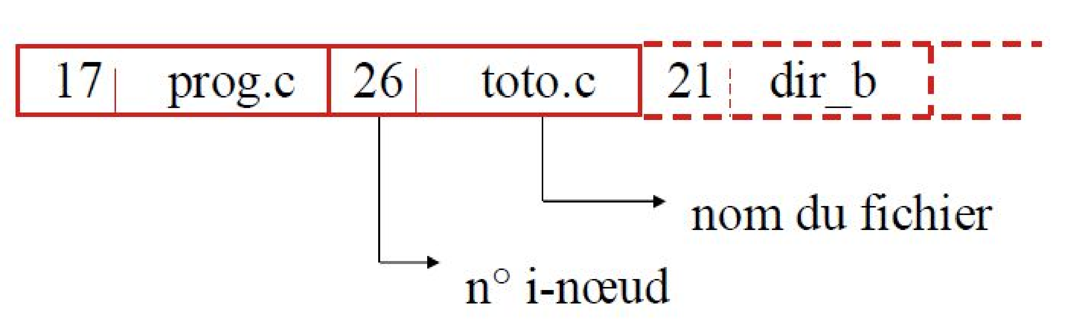
- *File system identification*
- *Time of last mount or write operation*
- *File system size in terms of blocks*
- *Size of a block Size of i-node structure*
- *Number of free blocks and inodes*
- *Number of blocks and nodes per group*
- *no. of 1st usable block*
- *no. of 1st unallocated i-node*
- *Time and frequency of last consistency check and other additional information*

- Linux Ext2 descriptors
  - *Block group descriptors*
    - *One descriptor/group of blocks*
    - *The bitmap block numbers managing*
      - *the list of free blocks*
      - *the list of free i-nodes*
    - *The number of free blocks and free i-nodes*
    - *The number of directories in the group*
    - *no. of the 1st block in the inode table*
  - *Bitmap*
    - *vectors to manage*
      - *The list of free i-nodes*
      - *The list of free blocks*
      - *0/ free 1/ occupied*

Descriptors are duplicated in each group of blocks

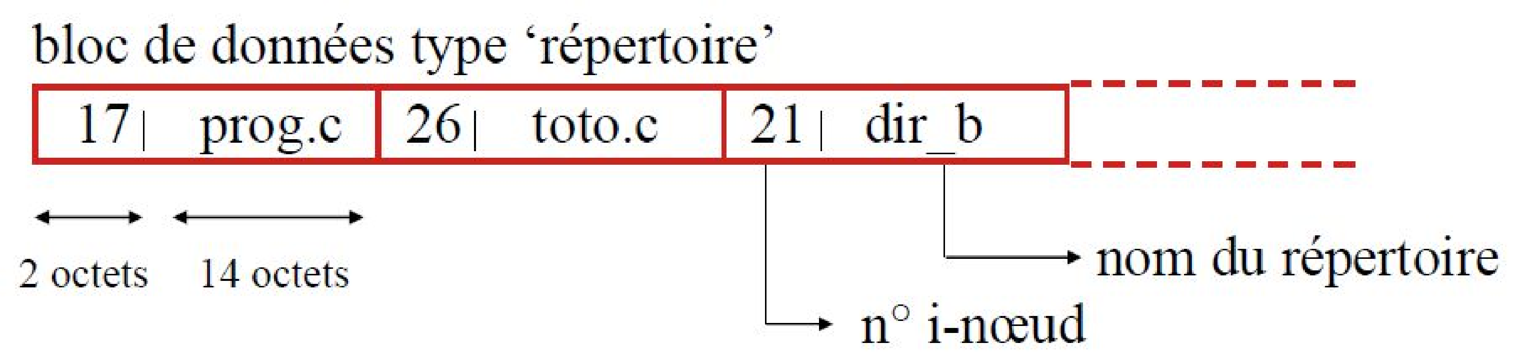
❑Notion of file

- *Identification: file name*
- *Locating the file using an algorithm ('namei')*
- *Browse 'directory'-type data blocks*
- *Search of the file name in the directory*



- *i-node of the file in the table of i-nodes*
  - ➢ *pointers to data blocks*

- Notion of directory
  - *"d" type file referenced in the i-nodes table*
    - *Points to 'directory' type data blocks*
    - *Directory data block = n * 16 bytes*
    - *no. of i-node*
    - *Name of file or directory*
  - *Root /: i-node no. 2*



- The 'namei' algorithm calls other algorithms:
  - *alloc' reserve file space on disk*
  - *free' free file space*
  - *ialloc' i-node reservation for a file*
  - *ifree' free i-node for a file*

i-node (index node): inode

- Notion i-node
  - *This is a data structure containing information about a file or directory stored in certain file systems.*
  - *Each file has a corresponding inode number in the file system in which it resides, unique to the device on which it is located.*
  - *To make it easier to handle data and file names, and to ensure a uniform file system structure, data is identified internally not by a name, but by a number, the i-node, which is independent of the actual address of the data on the disk.*
  - *The i-node provides an interface between the user and the physical file system.*

❑ i-node on disk

- *A file is characterized by an i-node which points to a block of data*
- *Table of i-nodes is located after the super-block*
- *An i-node number is the entry number in the disk i-node table*
- *Information contained in an i-node (64 bytes):*
  - *Owner name*
  - *Type of file*
  - *Permissions (access rights)*
  - *Date of last access*
  - *Number of links to other files (directory)*
  - *Size: number of bytes*
  - *Table of data block addresses*

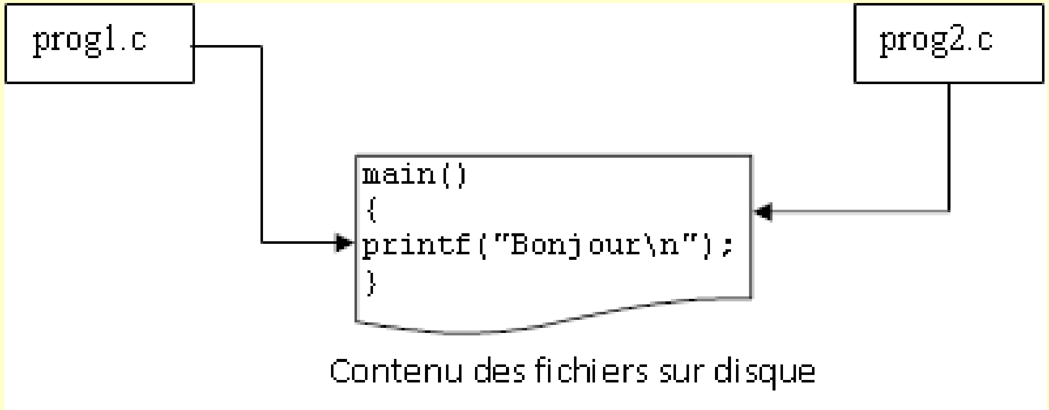
Physical and symbolic links

- They avoid duplication (content in several places).
- They ensure that updates are consistent, i.e. any update to the content of the source file will be displayed in the other files.
- There are two types of links used to redirect one file to another.
  - *Physical links*
  - *Symbolic links*

Physical links

- They allow two or more files to be associated with the same space on the disk.
- This operation is performed using the command ***ln***.
- **Example**:

```
ln prog1.c /home/etudiant22/prog2.c
```



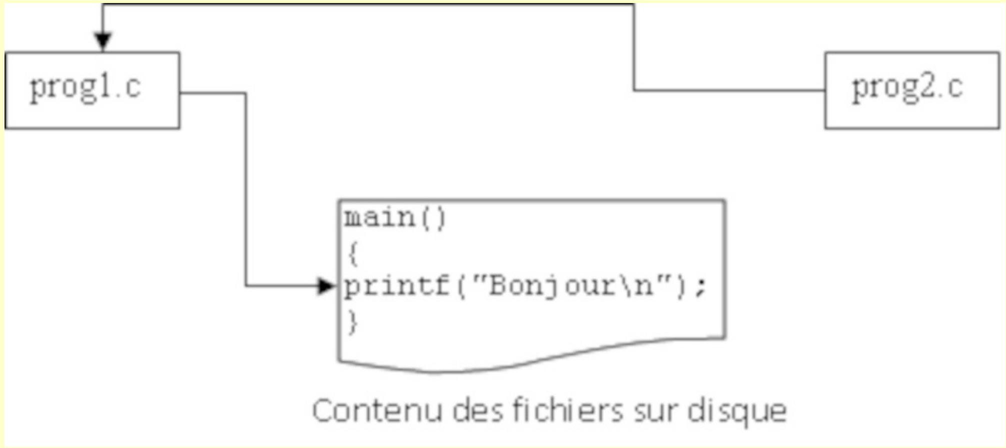
```
main()
{
printf("Bonjour\n");
}
```

- You can see that these 2 files are the same size.
- In terms of management, they are independent, but share the same disk space.
- Any modification to one modifies the other!
  - *But removing one breaks the link, but does not physically remove the other.*

Symbolic links

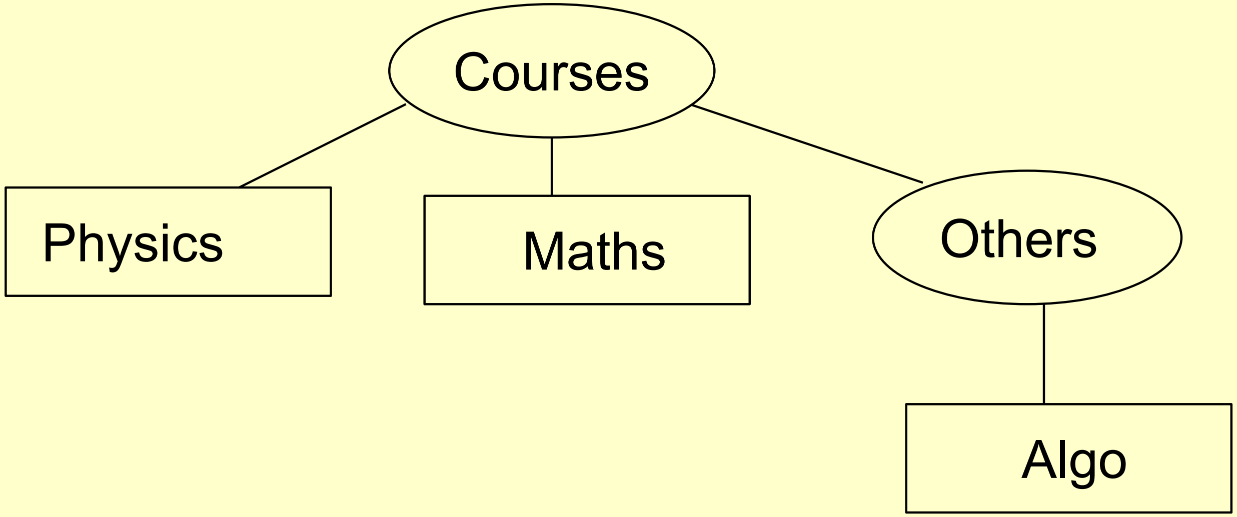
- ❑ The symbolic link refers to a file, and removing the source file will change the behavior of the link file, which will be considered 'broken’.
- ❑ This operation is performed using the command ***ln -s***.
- ❑ **Example**:

```
ln -s prog1.c /home/etudiant22/prog2.c
```



Example

- Consider the following tree structure:



  - ***Courses*** *and* ***Others*** *are directories*
  - ***Physics****,* ***Maths*** *and* ***Algo*** *are files*

- Examples of displaying i-nodes
  - *This is done with the command :*

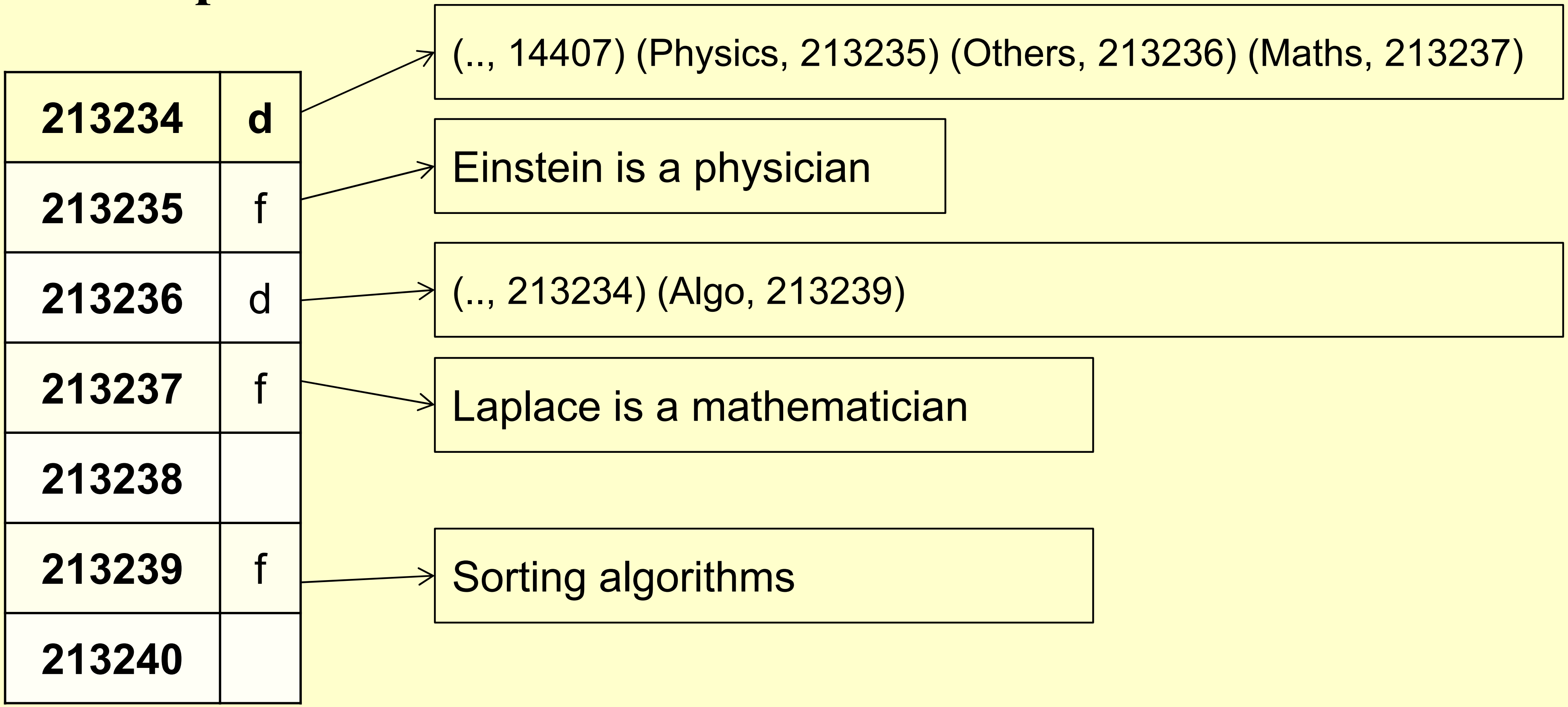
`ls -ilR` *(run this command in the directory Courses)*

```
213235  -rw-r--r--  1  Mohamed  Engineer 10 mar 20 15:57   PhysiCS
213237  -rw-r--r--  1  Mohamed  Engineer 10 mar 20 15:58   Maths
213236  drwxr-xr-x  2  Mohamed  Engineer 1024 mar 20 15:58 Others

Others:
213239  -rw-r--r--  1 Mohamed  engineer 12 mar 20 15:58 Algo
```

- *An i-node table contains a pointer to the stored information for each number; it also contains information about the file type, rights, owner and group, link count, modification date, etc.*
- *Directory information contains, for each element it contains, the number of the corresponding i-node, and that of its parent.*

❑ Example of inode table :



213234	**d**
213235	f
213236	d
213237	f
213238	
213239	f
213240	

(.., 14407) (Physics, 213235) (Others, 213236) (Maths, 213237)

Einstein is a physician

(.., 213234) (Algo, 213239)

Laplace is a mathematician

Sorting algorithms

- *In this way, it is possible to change the information without changing the name, or to change the name without changing the information, and to create, remove, rename and link files and directories.*

❑ Effects of commands : cp, mv, ln, rm

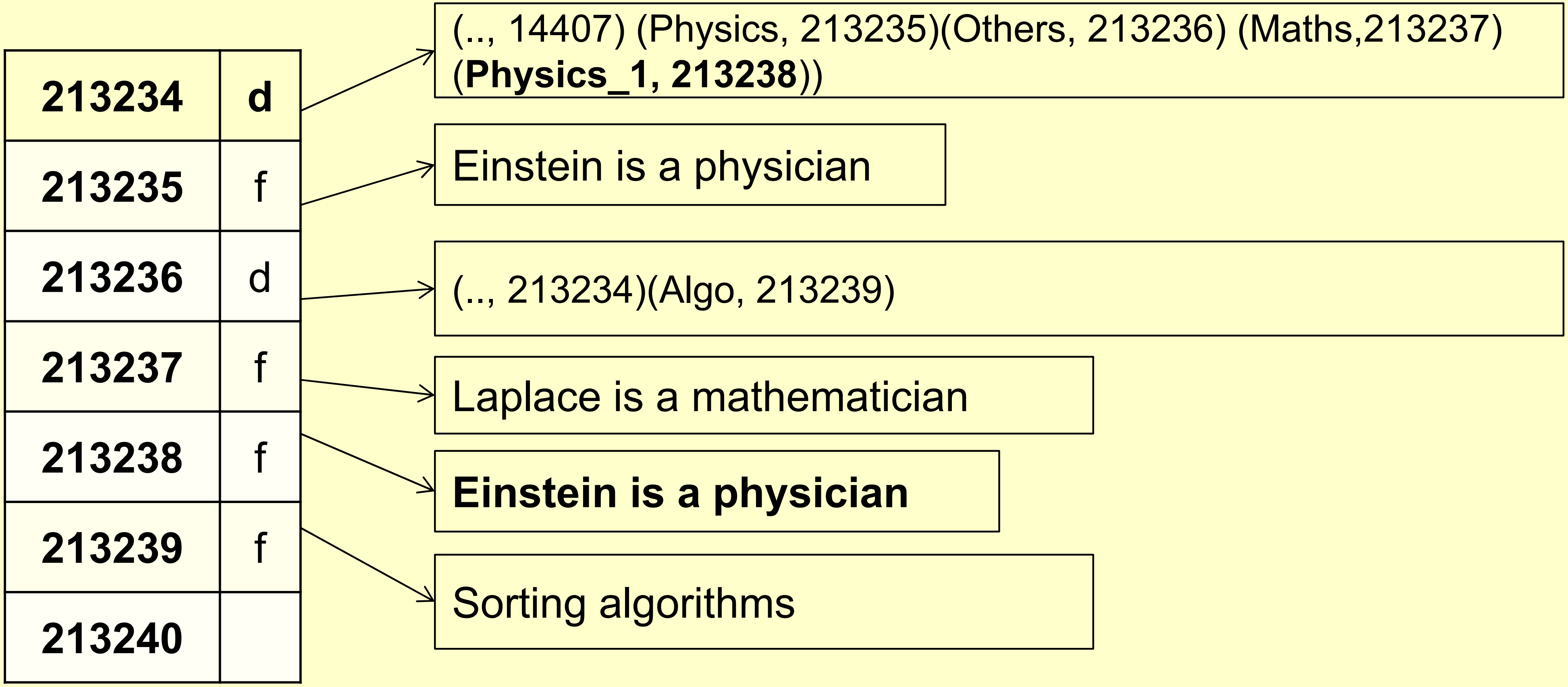
```
cp physics Physics_1
ln Maths Algebra (Links the file Maths to Algebra)
ln Maths Others (Creates a link Maths in the directory
  Others)
mv Maths Maths_2
```

❑ File system status :

```
213237 -rw-r--r-- 3 Mohamed  Group1 10 mar 20 15:58 Algebra
213235 -rw-r--r-- 1 Mohamed  Group1 10 mar 20 15:57 Physics
213237 -rw-r--r-- 3 Mohamed  Group1 10 mar 20 15:58 Maths_2
213236 drwxr-xr-x 2 Mohamed  Group1 10 mar 20 16:18 Others
213238 -rw-r--r-- 1 Mohamed  Group1 10 mar 20 16:18 Physics_1

213239 -rw-r--r-- 1 Mohamed  Group1 12 mar 20 15:58 Algo
213237 -rw-r--r-- 3 Mohamed  Group1 10 mar 20 15:58 Maths
```

- *Effects of command :* `cp   Physics Physics_1`



213234	**d**
213235	f
213236	d
213237	f
213238	f
213239	f
213240	

(.., 14407) (Physics, 213235)(Others, 213236) (Maths,213237) (**Physics_1, 213238**))

Einstein is a physician

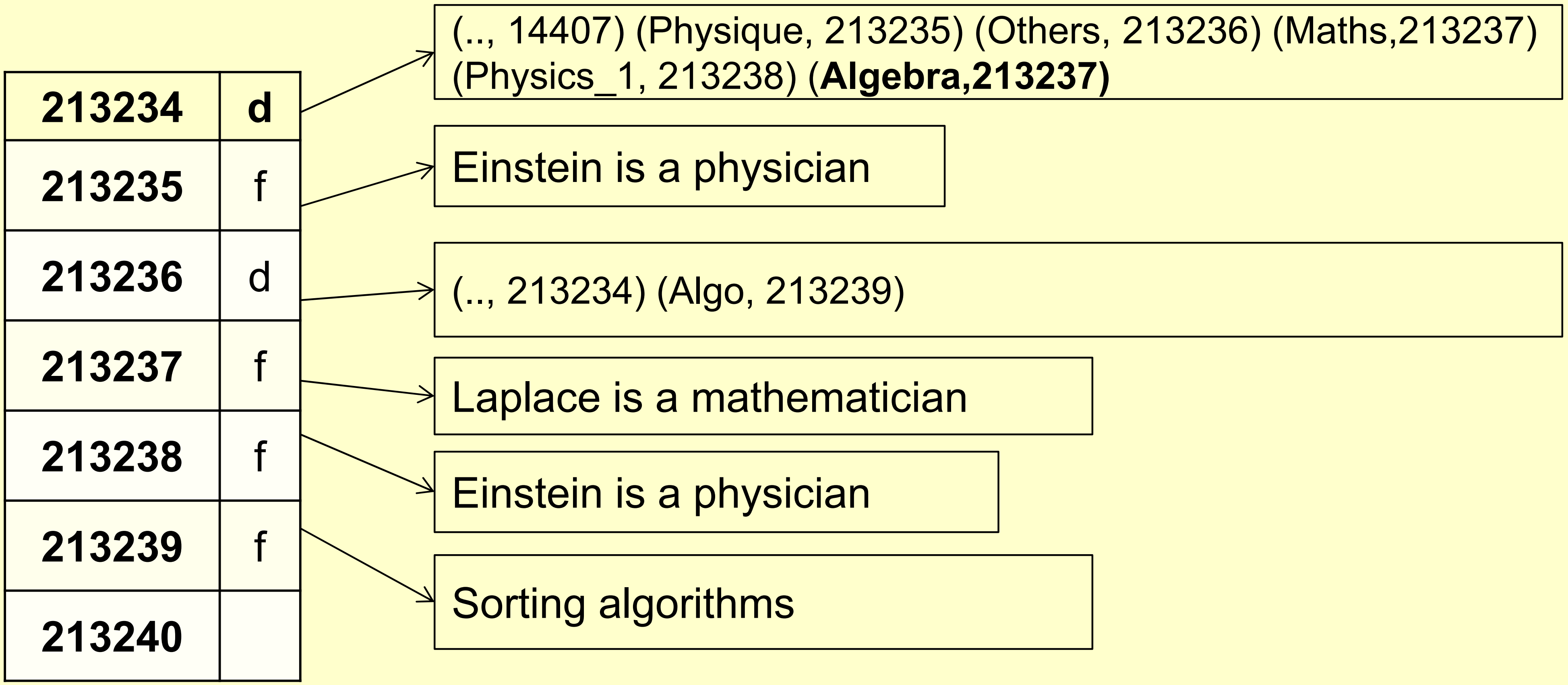
(.., 213234)(Algo, 213239)

Laplace is a mathematician

Einstein is a physician

Sorting algorithms

- *Effect of the command :* ***ln Maths Algebra***



213234	**d**
213235	f
213236	d
213237	f
213238	f
213239	f
213240	

(.., 14407) (Physique, 213235) (Others, 213236) (Maths,213237) (Physics_1, 213238) (**Algebra,213237)**

Einstein is a physician

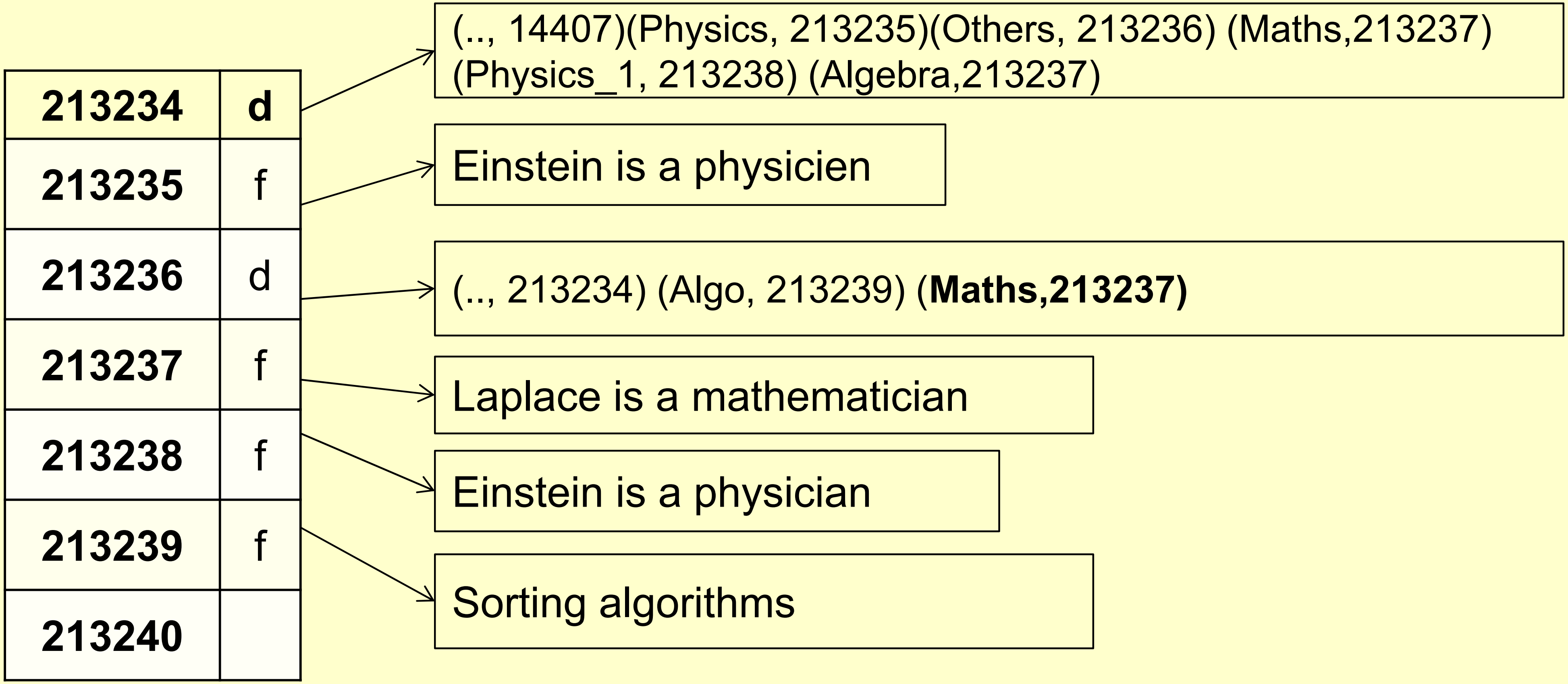
(.., 213234) (Algo, 213239)

Laplace is a mathematician

Einstein is a physician

Sorting algorithms

- *Effect of the command :* ***ln Maths Others***



213234	**d**
213235	f
213236	d
213237	f
213238	f
213239	f
213240	

(.., 14407)(Physics, 213235)(Others, 213236) (Maths,213237) (Physics_1, 213238) (Algebra,213237)

Einstein is a physicien

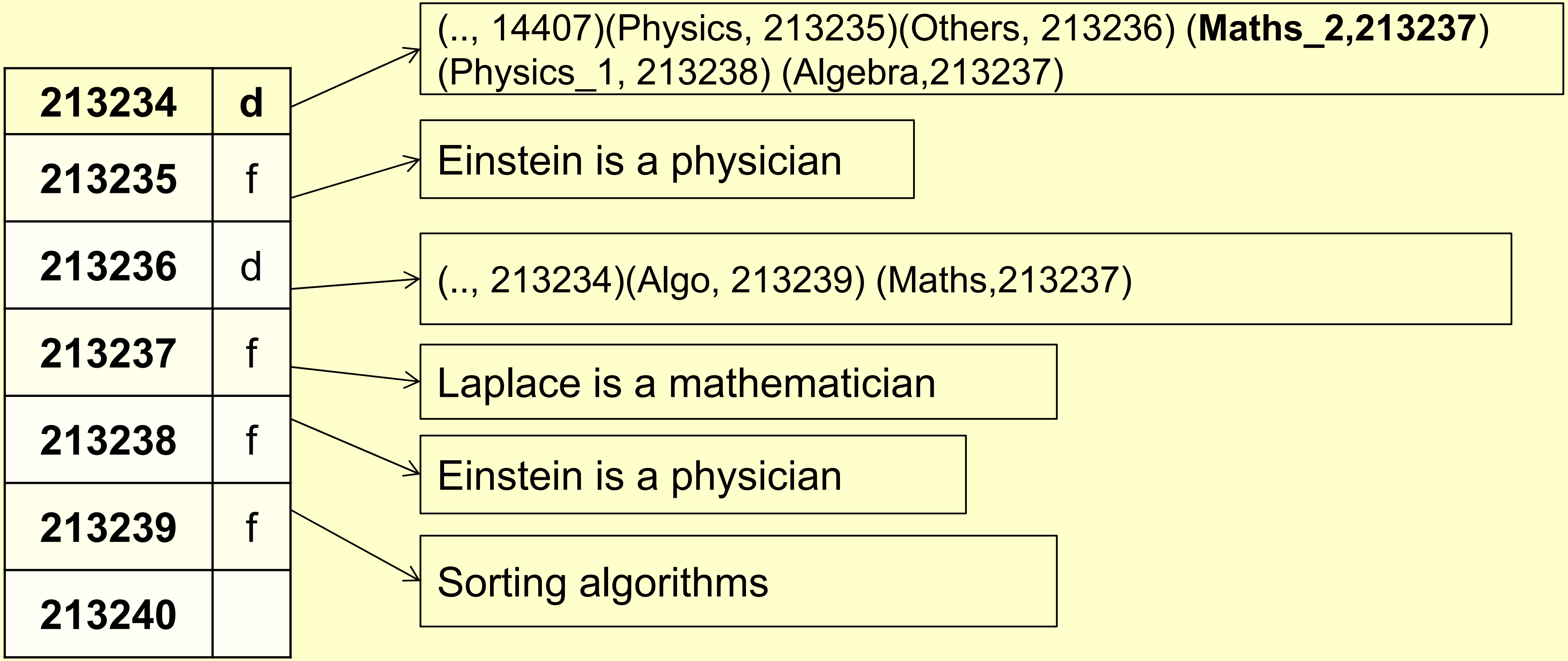
(.., 213234) (Algo, 213239) **(Maths,213237)**

Laplace is a mathematician

Einstein is a physician

Sorting algorithms

- *Effect of the command :* ***mv Maths Maths_2***



213234	**d**
213235	f
213236	d
213237	f
213238	f
213239	f
213240	

(.., 14407)(Physics, 213235)(Others, 213236) (**Maths_2,213237**) (Physics_1, 213238) (Algebra,213237)

Einstein is a physician

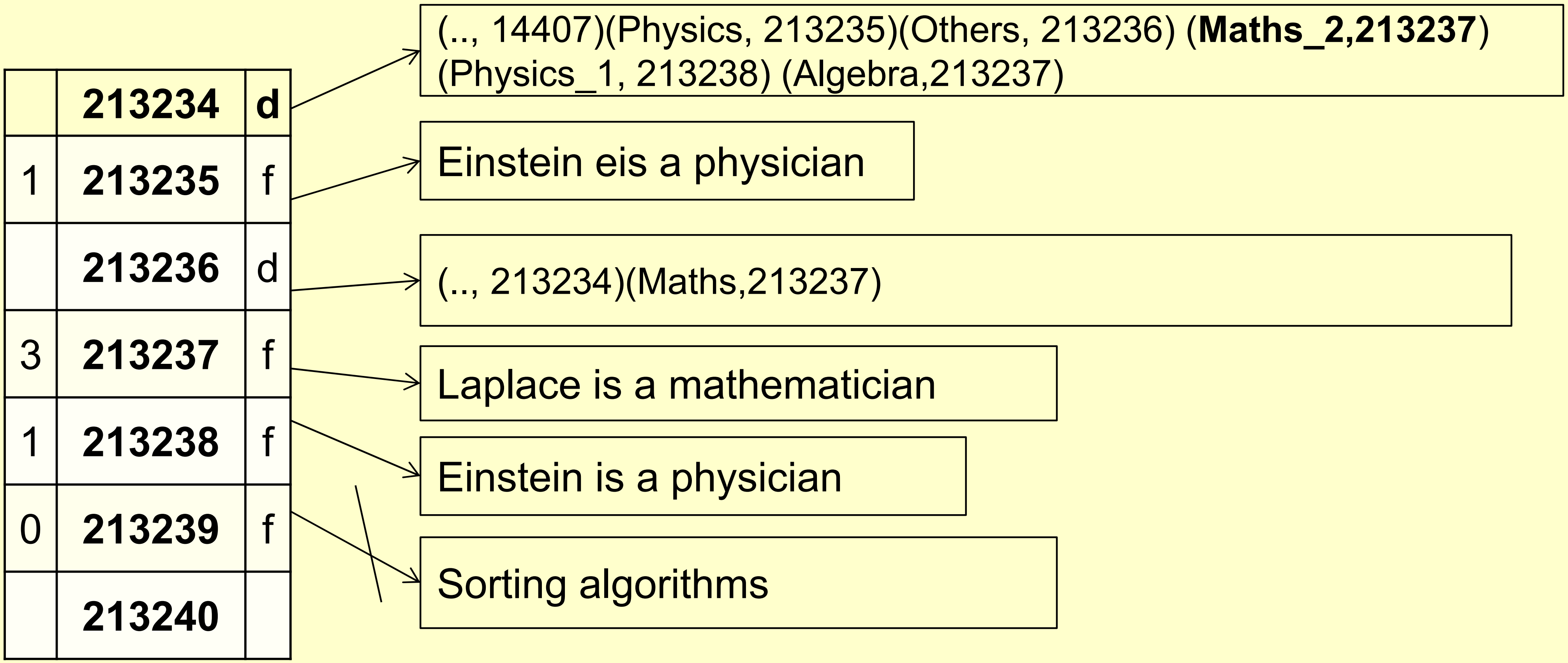
(.., 213234)(Algo, 213239) (Maths,213237)

Laplace is a mathematician

Einstein is a physician

Sorting algorithms

- *Effect of the command : **rm Others/Algo***



	213234	**d**
1	**213235**	f
	213236	d
3	**213237**	f
1	**213238**	f
0	**213239**	f
	213240	

(.., 14407)(Physics, 213235)(Others, 213236) (**Maths_2,213237**) (Physics_1, 213238) (Algebra,213237)

Einstein eis a physician

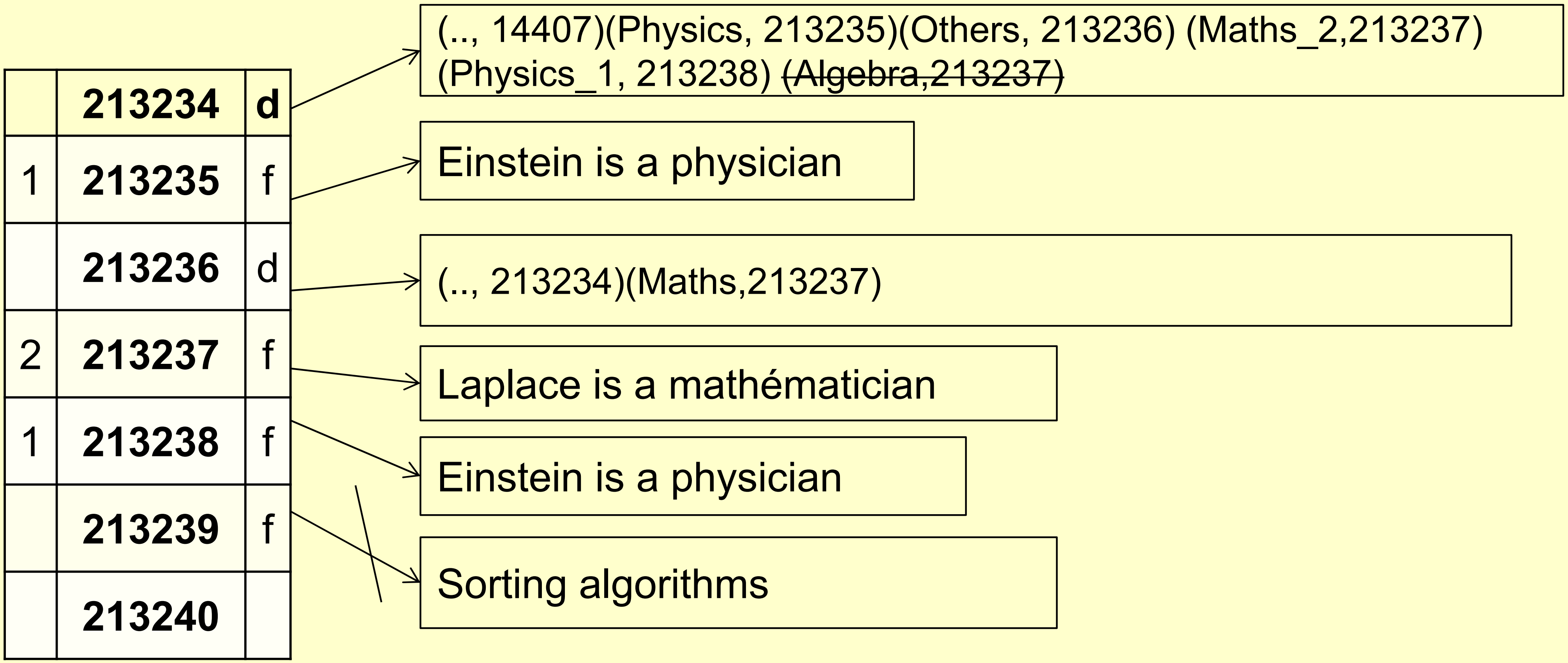
(.., 213234)(Maths,213237)

Laplace is a mathematician

Einstein is a physician

Sorting algorithms

- *Effet de la commande :* ***rm Algebra***



	213234	**d**
1	**213235**	f
	213236	d
2	**213237**	f
1	**213238**	f
	213239	f
	213240	

(.., 14407)(Physics, 213235)(Others, 213236) (Maths_2,213237) (Physics_1, 213238) ~~(Algebra,213237)~~

Einstein is a physician

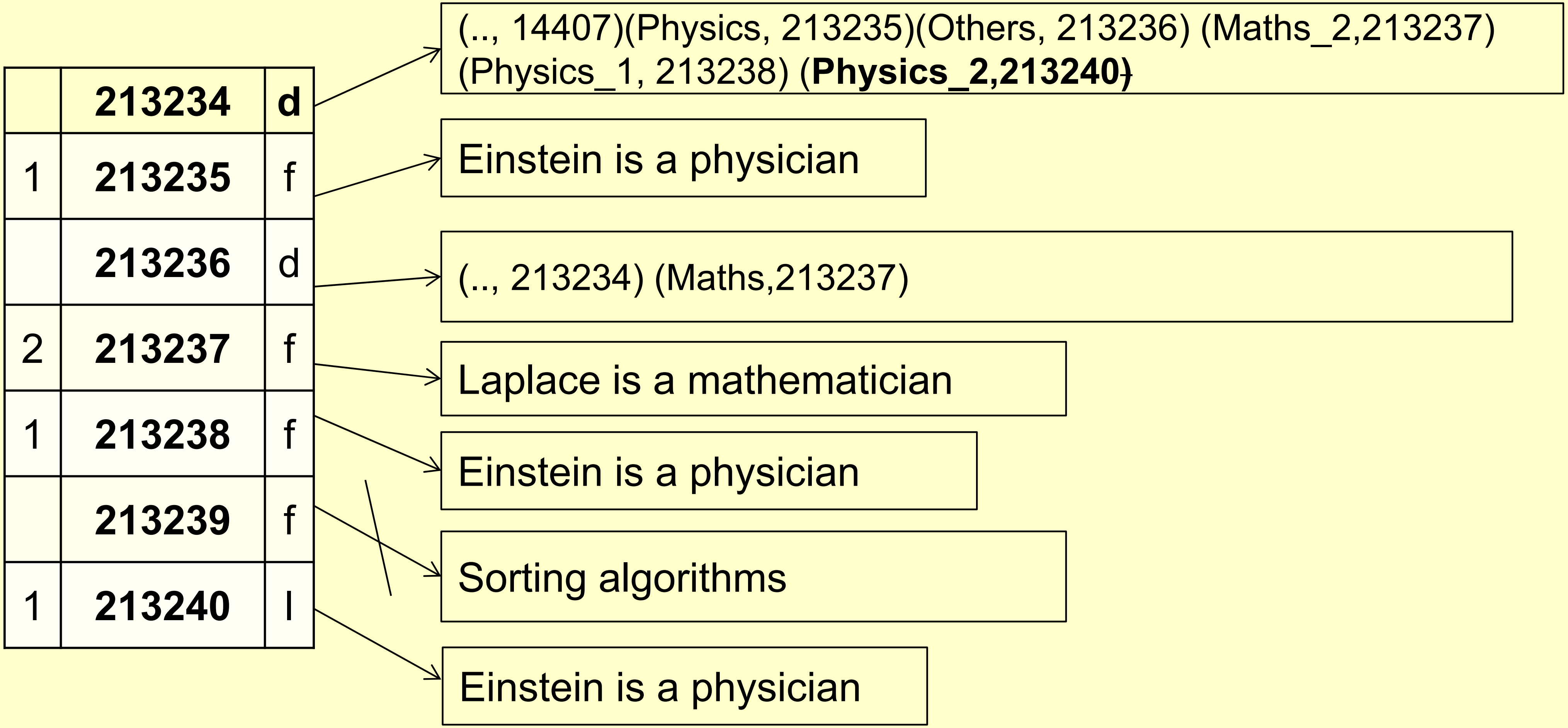
(.., 213234)(Maths,213237)

Laplace is a mathématician

Einstein is a physician

Sorting algorithms

- *Effet de la commande :* ***ln –s physics Physics_2***



	213234	**d**
1	**213235**	f
	213236	d
2	**213237**	f
1	**213238**	f
	213239	f
1	**213240**	l

(.., 14407)(Physics, 213235)(Others, 213236) (Maths_2,213237) (Physics_1, 213238) (**Physics_2,213240**)

Einstein is a physician

(.., 213234) (Maths,213237)

Laplace is a mathematician

Einstein is a physician

Sorting algorithms

Einstein is a physician

File organization on disk

- Organization
  - *Non-contiguous (blocks of data in a file do not appear one after the other)*
  - *Dynamic disk space management (as required)*
- Blocks
  - *Are the same size*
  - *There are two types of block*
    - *data blocks*
    - *address blocks: addresses to data blocks or to address blocks*
  - *Table of i-nodes on disk uses these different types of blocks*

- How to choose block size
  - *Large size ---> reduces the number of blocks -- > loss of space due to occupancy rate*
  - *Small size ---> increases the number of blocks -- > wastes time searching in several places*

- For each i-node in the i-node table
  - *13 entries (addresses or pointers)*
    - *10 @ direct to data blocks (or 12 for Ext2)*
    - *3 indirect @:*
      - *single indirect block no.: block no. with 256 data block nos.*
      - *double indirect block no.: block no. with 256 address block nos.*
      - *triple indirect block no.: block no. with 256 address block no. address blocks address blocks*
  - *If address on 32 bits and block of 1024 bytes*

    => maximum file size 16 GB (in theory)
  - *a free block is zero*

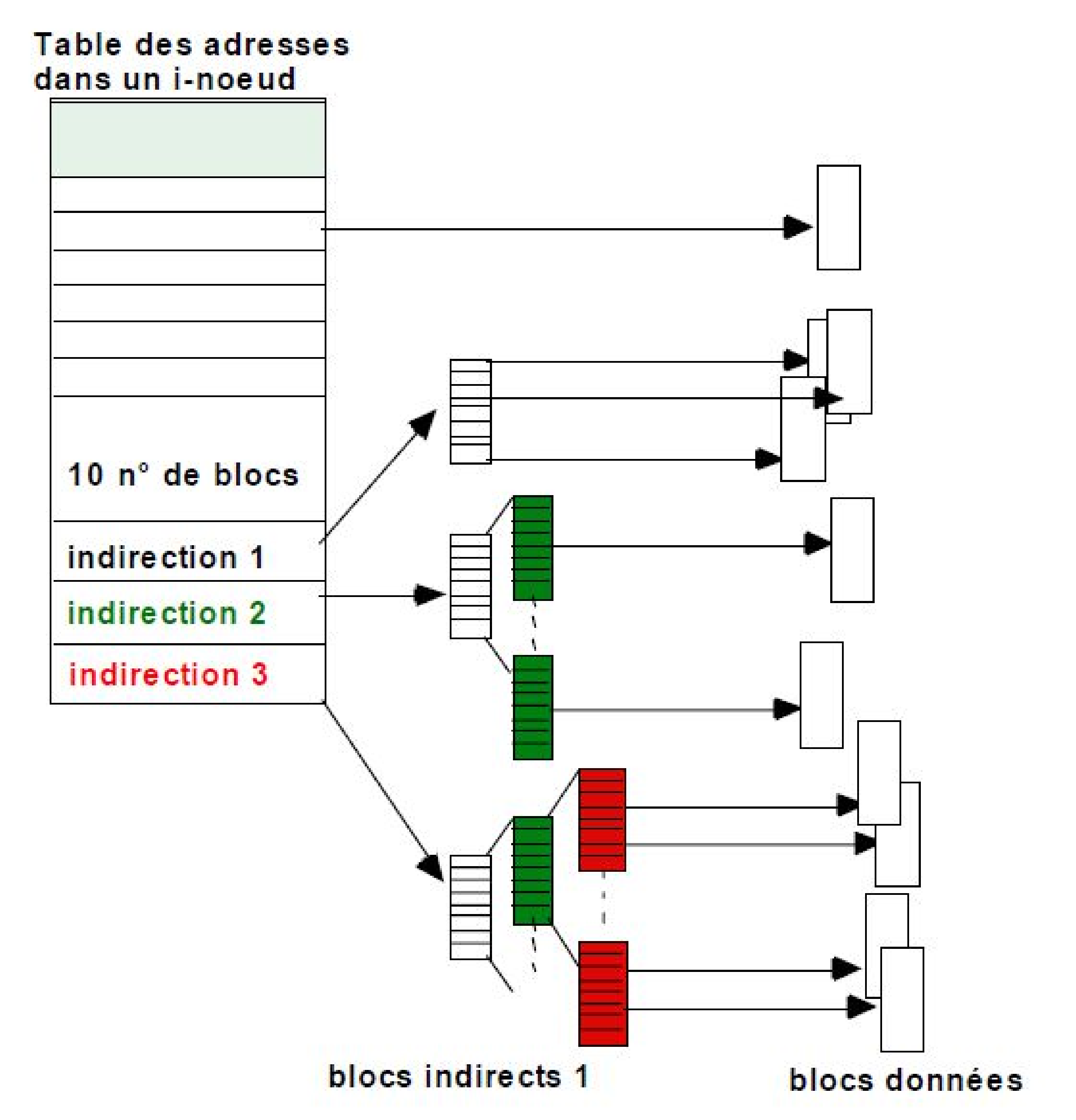


Table des adresses dans un i-noeud
10 n° de blocs
indirection 1
indirection 2
indirection 3
blocs indirects 1
blocs données

❑ Example:

- *How many blocks are needed to store a file with 1025 data blocks?*
  - ➢ *10 blocks of direct access data*
  - ➢ *256 single indirect access data blocks*
  - ➢ *(256 + 256 + 247) double-access data blocks*
  - ➢ *In total we have :*
  - ➢ *1025 data blocks*
  - ➢ *1 single indirect access address block*
  - ➢ *4 double indirect access address blocks*

1030 blocks