

# Fundamentos de los Sistemas Operativos

Departamento de Informática de Sistemas y Computadoras (DISCA)  
*Universitat Politècnica de València*



## **Lab session 7** **MINIX File System:** **Structure visualization**

### **Content**

1. Objectives .....	2
2. Previous steps.....	2
3. Superblock and root directory.....	2
4. Analyzing a directory structure .....	3
5. Exercise 3: Identifying file types .....	5
6. Annexes .....	8
6.1 Introduction to the Minix file system .....	8
6.2 vmfs tool handling .....	9
6.3 Experimenting on a Minix file system: sudo privileges required .....	11

## 1. Objectives

Analyzing, through visualization with Minix Viewer tool, the different parts of a MINIX file system. Successive images of a Minix file system, resulting from file operations, will be visualized and analyzed in order to study how the allocation of blocks to disk files is handled and to learn about the different data structures used to implement regular files, directories, and physical and symbolic links.

## 2. Previous steps

Download in \$HOME the following files from the lab folder in PoliformaT:

- jar file corresponding to the Minix file systems viewer: MinixViewer.jar
- The file system image files: minixfsXX. They correspond to a 12-megabyte partition and are the successive states of the file system, after performing the file operations mentioned in this guide. If you have a Linux system of your own, you can create these images as explained in annex 6.3. Root privileges required ("sudo" command).

## 3. Superblock and root directory

Start the viewer with the command:

```
java -jar MinixViewer.jar &
```

Next, on the viewer window, click on File and then Open, now select "minixfs00". This image corresponds to a newly created Minix file system.

**Question 1:** Write down the values that appear in "Super-block" section.

Number of i-nodes  
Number of zones  
1st data zone

Data in the superblock given by MinixViewer

**Question 2:** Answer the following questions accurately:

How many blocks occupies the i-node map?	
What i-nodes are occupied?	
What do these i-nodes correspond to?	
How many blocks occupies the zone map?	
How many blocks are occupied by the i-nodes?	
Look at "Zone map" and verify that the first block in the data zone is busy. What file and what i-node corresponds to this block?	

**Question 3:** Click on the i-nodes tab on the viewer, fill in the table for **i-node 1** and explain the values obtained:

	i-node 1
Mode	
UID	
Size	
Instant	
GID	
Links	
Zone 0	
Zone 1	
Zone 2	
Zone 3	
Zone 4	
Zone 5	
Zone 6	
Zone SI	
Zone DI	

Explain the value of field "Size"	
Explain the value of field "Links"	

#### 4. Analyzing a directory structure

On the working Minix file system it has been created the directory structure show on Figure 1, where *bin/lis* is a hard link to *users/alfonso/mils*, and *usr/prac2* is a symbolic link to *usr/prac1*

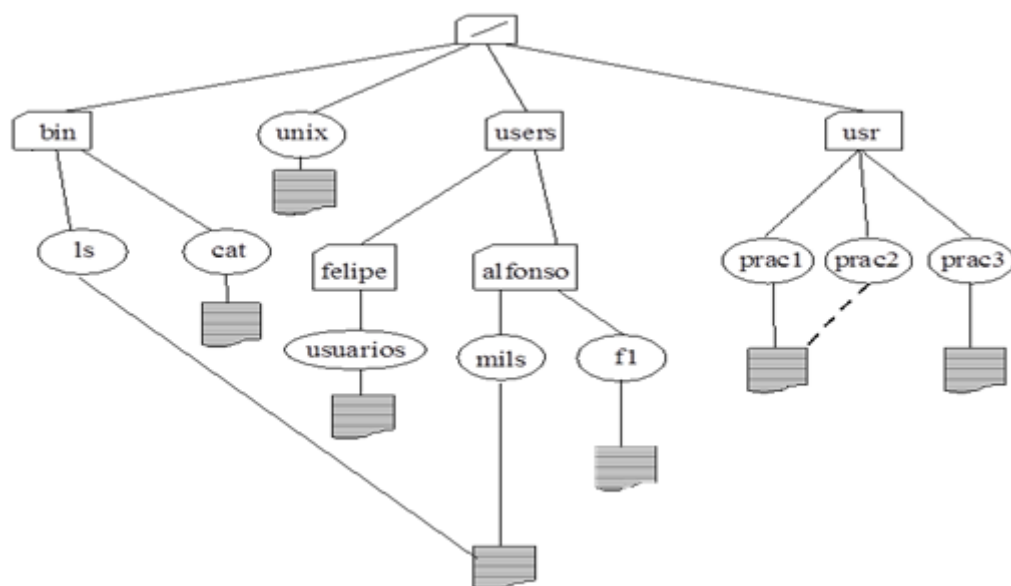


Figure 1: Directory structure created

**Go to the viewer window and load file “minixfs01”, which corresponds to the new state of the file system.**

**Question 4:** With the new structure created, answer the following questions:

Have the superblock structure data changed after the creation of the file structure? Whether they have changed or not, explain why.	
How many and which i-nodes are occupied?	
How many and what data blocks are occupied?	

**Question 5:** Fill the following table for i-node 1

i-node 1	
Mode	
UID	
Size	
Instant	
GID	
Links	
Zone 0	
Zone 1	
Zone 2	
Zone 3	
Zone 4	
Zone 5	
Zone 6	
Zone SI	
Zone DI	

**Question 6:** Explain the changes that appear in the values of its fields and compare them to the ones got previously.

## 5. Exercise 3: Identifying file types

The *stat* and *fstat* system calls provide information of a file by referring to it in its i-node. See the manual for these calls with command *man fstat*. Particularly, confirm the values (in octal) that identify the following types of files in the *st\_mode* field of *stat* structure:

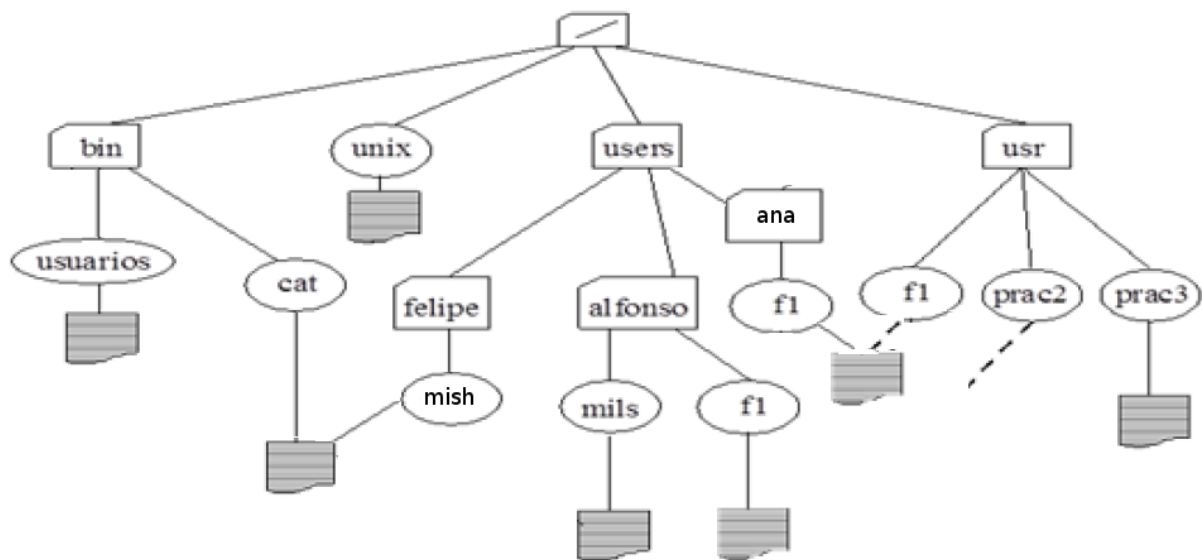
File type	Value
Regular	0100000
Directory	0040000
Symbolic link	0120000
FIFO	0010000

**Question 7:** Look at the 'Mode' field that you have filled in question 5. What kind of file corresponds to that value? Check the 'Mode' value is consistent with the type of file and the permission bits associated to the permissions word. Indicate the meaning of the position of the digits in the 'Mode' field.

From the shell the following commands were executed, with the minix root directory as the working directory :

```
mkdir users/ana
cp users/alfonso/f1 users/ana
mv users/felipe/usuarios bin
rm bin/ls
rm usr/prac1
ln bin/cat users/felipe/mish
ln -s users/ana/f1 usr/f1
```

The following image depicts the file system state, as it is in minixfs02:



**Question 8: On the viewer load file minixfs02.** Go to the “File/Directory content” tab and display the contents of files *users/felipe/mish* and *usr/f1*, and compare them to the *bin/cat* and *users/ana/f1*, annotating ASCII content of the first line on the following table.

**Note.** File names are relative to Minix file system root directory (\$HOME/minix)

<b>users/felipe/mish</b>	
<b>usr/f1</b>	
<b>bin/cat</b>	
<b>users/ana/f1</b>	

**Question 9:** Relying on the information provided by Minix Viewer, fill in the following table:

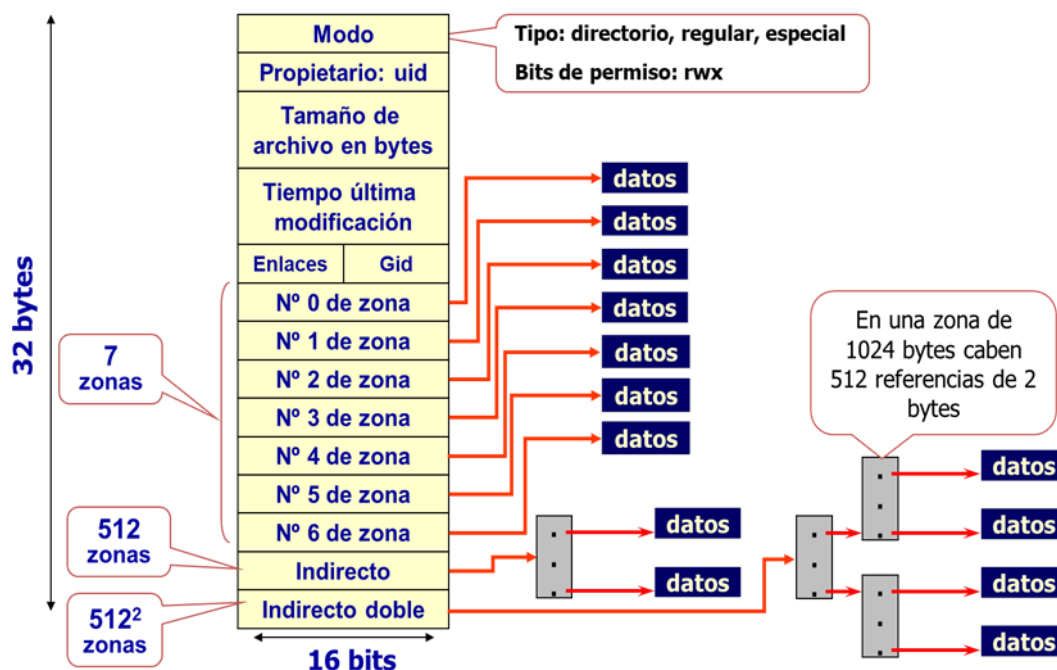
	<b>usr/prac3</b>	<b>bin/usuarios</b>	<b>unix</b>
Type of addressing modes to zones are used in the i-node: direct, indirect, double indirect			
Number of blocks occupied with references to block			
Size in bytes of the file			
Number of blocks occupied with own file data			
Total number of blocks allocated			

## 6. Annexes

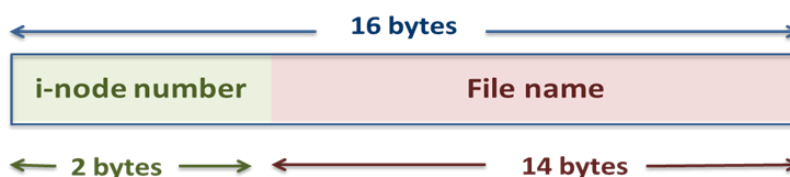
### 6.1 Introduction to the Minix file system

The MINIX file system is characterized by:

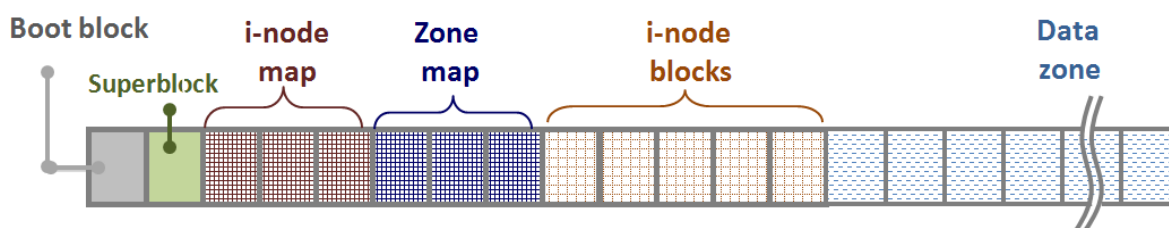
- Using the method of indexed allocation in a variant of combined schemes to allocate space to files.
- Managing the free disk space by using bitmap, with a 0 in bit  $i$  indicates that data  $i$  block is taken and a 1 indicates that the data  $i$  block is available.
- Saving file attributes, with the exception of the name, in special structures called i-nodes, consisting of 32 bytes and located physically on the disk out of the space allocated to the file.



- Directories entries are 16 bytes and contain the number of i-node and file name.



The structure of a MINIX file system involves interpreting the contents of a disk partition. The device manager provides an interface to the operating system, in which this partition is seen as a vector of fixed size fields.





The superblock contains the description of the structure of the partition like its size and location of its elements.

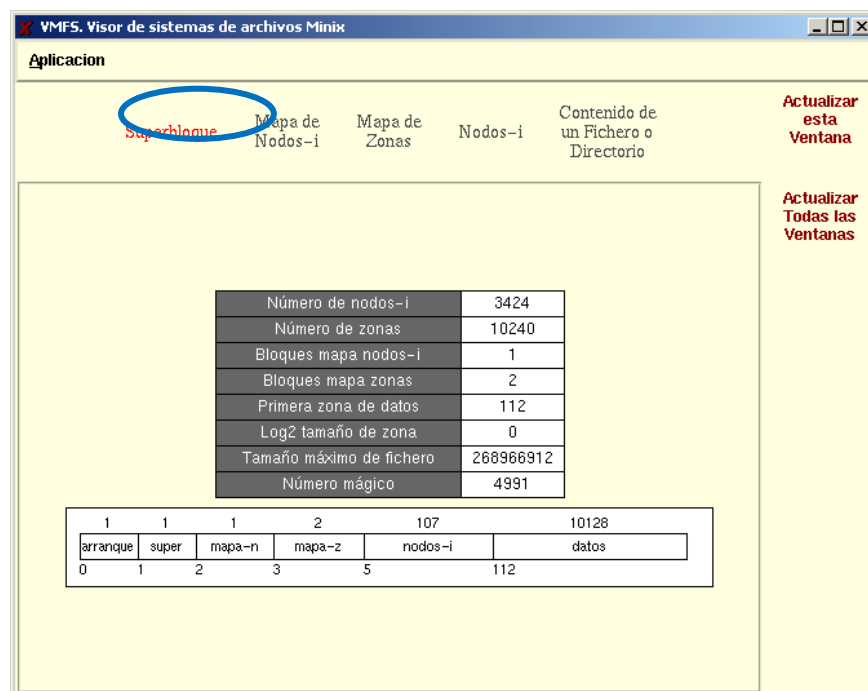
Superblock	Blocks number	It is fixed at partition creation, it can be less than partition size
	i-nodes number	It is fixed by the user at partition creation or it takes default values
	N (1 zone = $2^N$ blocks)	N value is stored into the superblock
	i-node map blocks number	$[i\text{-nodes number} / \text{Block bits number}]$
	Zone map blocks number	$[\text{Data zones number} / \text{Block bits number}]$
	First data block location	$2 \text{ (boot and superblock)} + i\text{-node map blocks number} + \text{zone map blocks number} + i\text{-node blocks number}$
	Magic number	Numerical value that guaranties that the partition contains a MINIX file system

## 6.2 vmfs tool handling

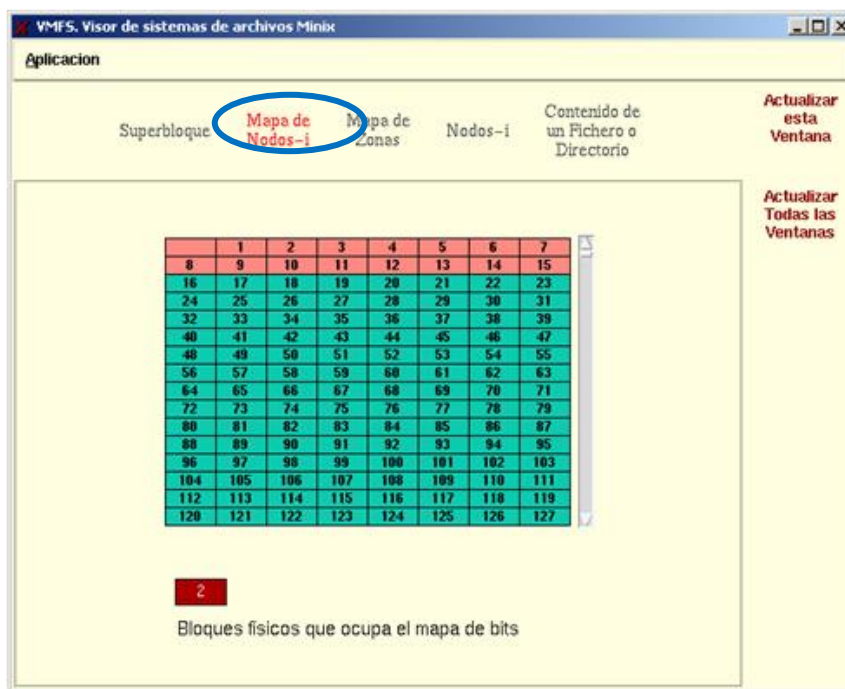
**Note.** This corresponds to the former application version made with Tcl/Tk. The new version is made on Java but it has basically the same functionality.

The tool vmfs (viewer of minix file system) was implemented with pedagogical purposes only. vmfs reads the blocks of a device containing a file system with Minix (v1) format, and graphically displays information about it. Allows viewing the contents of the superblock, and blocks that contain bitmaps, i-nodes, directories and file data..

The first information displayed is related to the superblock. The vmfs tool provides a screen with the following appearance:

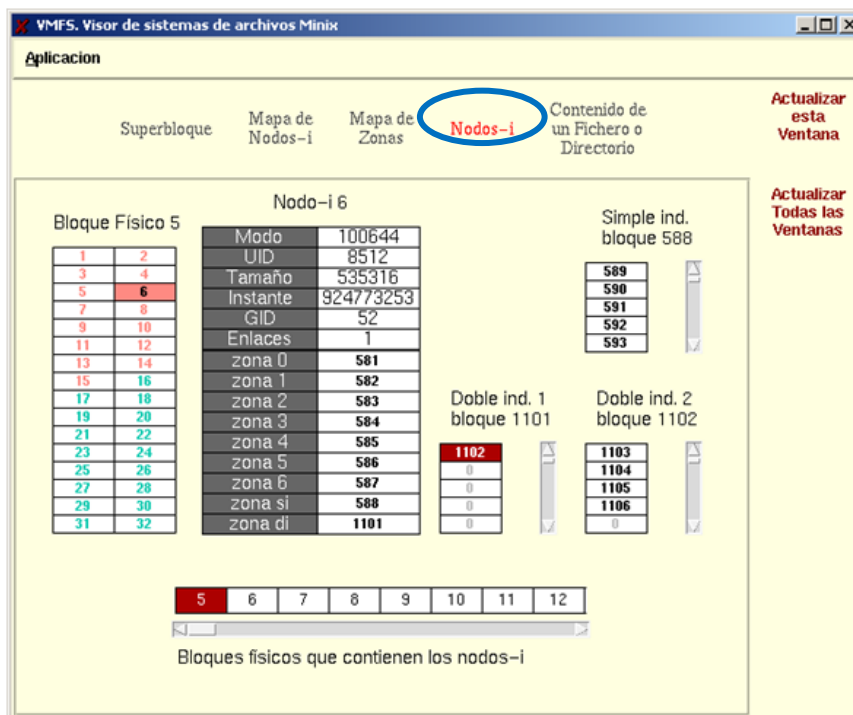


The different structures of a MINIX file system can be consulted by selecting options from the top menu. The i-node looks like this:



where the occupied i-nodes are colored in red and the free ones in green. Something similar is displayed if you select the zone map in the menu.

To view the structure of a i-node you only have to press the appropriate option from the menu.



### 6.3 Experimenting on a Minix file system: sudo privileges required

1. Verify that you have a JRE installed with command:

```
java -version
```

If it is not then install Java from your Linux distribution package manager.

2. Prepare a Minix file system with with the following commands:

```
cd
dd if=/dev/zero of=minixfs bs=1024 count=12288
mkfs -t minix -n 14 minixfs
mkdir minix
sudo mount -o loop=/dev/loop0, sync, uid=$(id -u), gid=$(id -g) minixfs minix
```

3. Download in \$HOME the jar file corresponding the the viewer *MinixViewer.jar*, contained in the lab folder in PoliformaT. Verify that it works with command:

```
java -jar MinixViewer.jar &
```

Now load file \$HOME/minixfs, with File -> Open, and analyze the superblock and root directory content, you will find the same as in section 3 on this lab guide.

4. Download in \$HOME file “ejemplo.tar”, contained on subfolder “src” in PoliformaT, next execute the following commands:

```
cd $HOME/minix
tar xvf ../ejemplo.tar
free && sync
```

5. Go to the viewer Windows and load file \$HOME/minixfs with File -> Open, you will find the same as in section 4 on this lab guide.

6. Download in \$HOME file “ordenes.sh”, contained on subfolder “src” in PoliformaT, and set execution permissions with command:

```
chmod 755 $HOME/ordenes.sh
```

Next execute the following commands:

```
cd $HOME/minix
../ordenes.sh
free && sync
```

7. Go to the viewer Windows and load file \$HOME/minixfs, with File -> Open, you will find the same as in section 5 on this lab guide.

Repeat this cycle “change + view” at your will. Commands “df” and “ls”, with the available options, will allow you to check the information shown on the viewer.