Fundamentos de los Sistemas Operativos

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

fSO

Lab session 7
MINIX File System:
Structure visualization

Content

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.

	Data in the superblock given by MinixViewer
Number of i-nodes	4096
Number of zones	12288
1st data zone	133

Question 2: Answer the following questions accurately:

How many blocks occupies the i-node bitmap?	1
What i-nodes are occupied? Which are these i-nodes?	Inode 1, which refers to the root (/) directory, is occupied
How many blocks occupies the zone bitmap?	2
How many blocks are occupied by the i-nodes?	128 blocks (from 5 to 132)
Look at "Zone map" and verify that the block first block in the data zone is busy. What file and what i-node corresponds to this block?	The first block in the data zone (block 133) is busy. It contains the root directory (/), and it's linked to inode 1.

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

	i-node 1
Mode	40755
UID	1000
Size	32
Instant	1544923363
GID	232
Links	2
Zone 0	133
Zone 1	0
Zone 2	0
Zone 3	0
Zone 4	0
Zone 5	0
Zone 6	0
Zone SI	0
Zone DI	0

Explain the value of field “Size”	In MINIX, directory entries take 16 bytes. Inode 1 points to root, which contians 2 directory entries. 2 * 16 = 32.
Explain the value of field “Links”	There are 2 links to the inode, and both are in the root directory. One refers to the current directory (.), and the other to the previous directory (..), which in the case of / is also the root directory.

4. Analyzing a directory structure

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

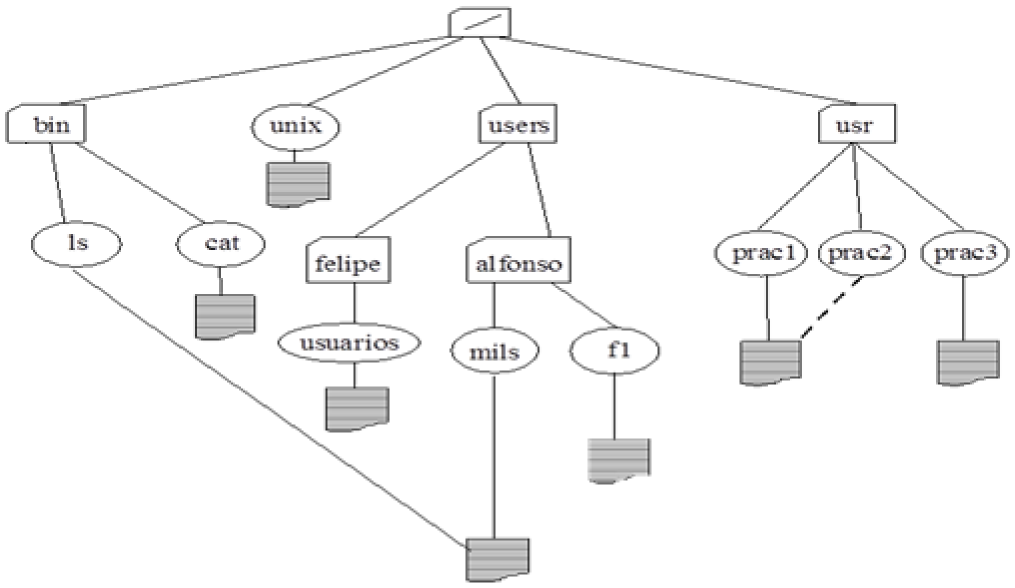


Figure 1: Directory structure created

Go to the viewer window and load file "minixfs01", that corresponds to the new state on the file system, and answer the following questions:

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

Have they changed the superblock data after you create the file structure? Whether they have changed or not explain why.	No. The superblock contains information about the structure of the filesystem, and it doesn't change when creating new files or directories.
How many and which nodos-i are occupied?	Now, inodes from 1 to 14 are occupied.
How many and what data blocks are occupied?	Zone blocks from 133 to 693 are occupied (561 in total).

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

	i-node 1
Mode	40755
UID	1000
Size	96
Instant	1545058285
GID	232
Links	5
Zone 0	133
Zone 1	0
Zone 2	0
Zone 3	0
Zone 4	0
Zone 5	0
Zone 6	0
Zone SI	0
Zone DI	0

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

The main changes are the amount of links and size. The size is calculated as 16 * file / directory entries. As now there are 3 directories + . + .. + a file (`unix`), the size is 16 * 6 = 96. 2 of the links to the inode are still in root (. and ..), and the other 3 are called `..`, and there is one inside each one of the 3 directories in / (`bin`, `users`, `usr`).

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 field *st_mode* of structure *stat*:

File type	**Value**
Regular	0**10**0000
Directory	00**4**0000
Symbolic link	0**12**0000
FIFO	00**1**0000

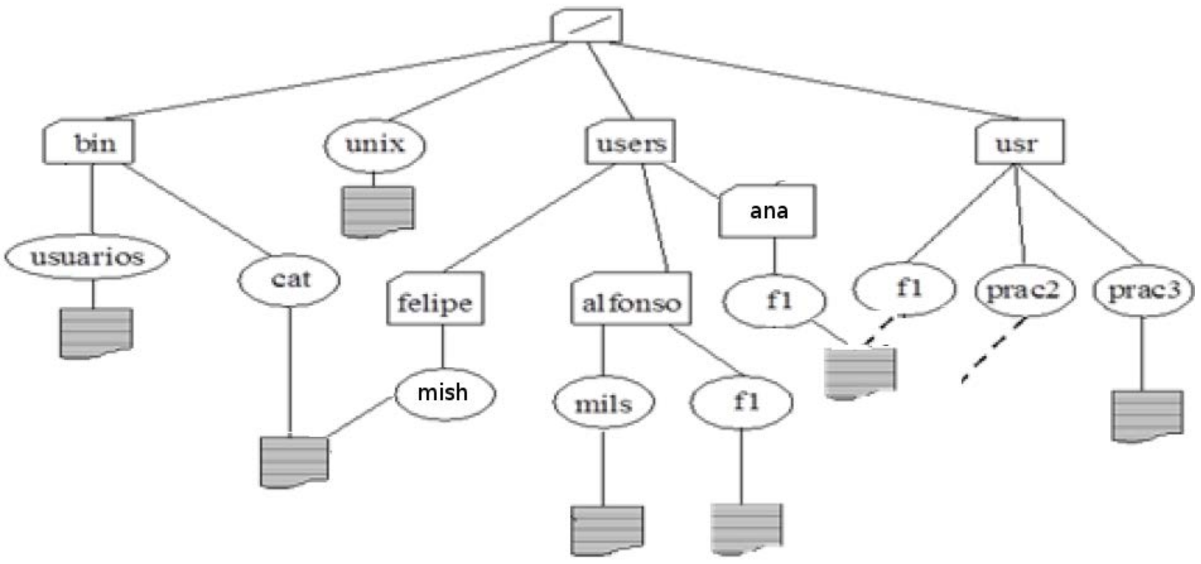
Question 7: Look at the field mode that you have filled in question 5, what kind of file corresponds to that value? Check the 'Mode' value is consistent with the type of file and the permission bits associated to the permissions word. Indicate the meaning of the position of the digits in the field "Mode".

The "Mode" value is 40755. 40 means it's a directory. 755 is the permission value of the file. It means everyone can read and execute it, but only the owner can write to it.

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)

users/felipe/mish	Hola, soy cat
usr/f1	users/ana/f1
bin/cat	Hola, soy cat
users/ana/f1	Hola, soy f1

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

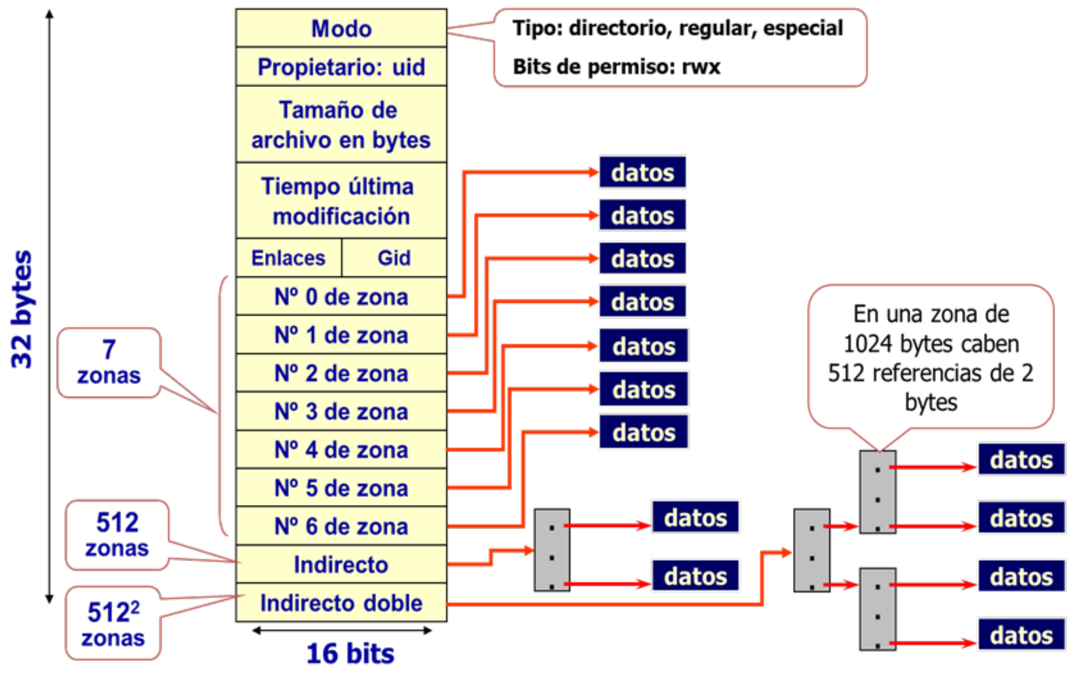
	usr/prac3	**bin/usuarios**	**unix**
Type of addressing modes to zones are used in the i-node: direct, indirect, double indirect	direct (zones 0 to 5)	direct (zone 0)	direct, indirect, and double indir.
Number of blocks occupied with references to block	0	0	1 + 512 = 513
Size in bytes of the file	5850	11	535316
Number of blocks occupied with own file data	6	1	ceil(535316 / 1024) = 523
Total number of blocks allocated	6	1	7 + 512 + 512^2 = 262663

6. Annexes

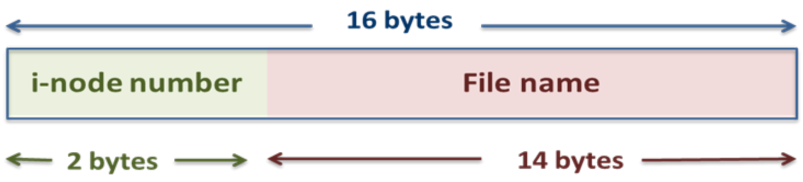
6.1 Introduction to the Minix file system

The MINIX file system is characterized by:

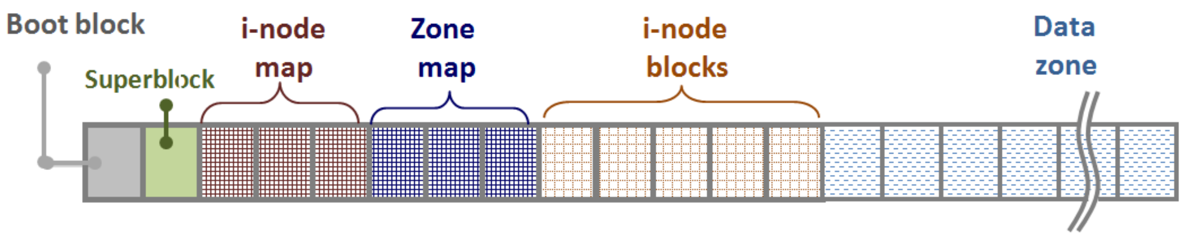
- Using the method of indexed allocation in a variant of combined schemes to allocate space to files.
- Managing the free disk space by using bitmap, with a 0 in bit i indicates that data i block is taken and a 1 indicates that the data i block is available.
- Saving file attributes, with the exception 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.



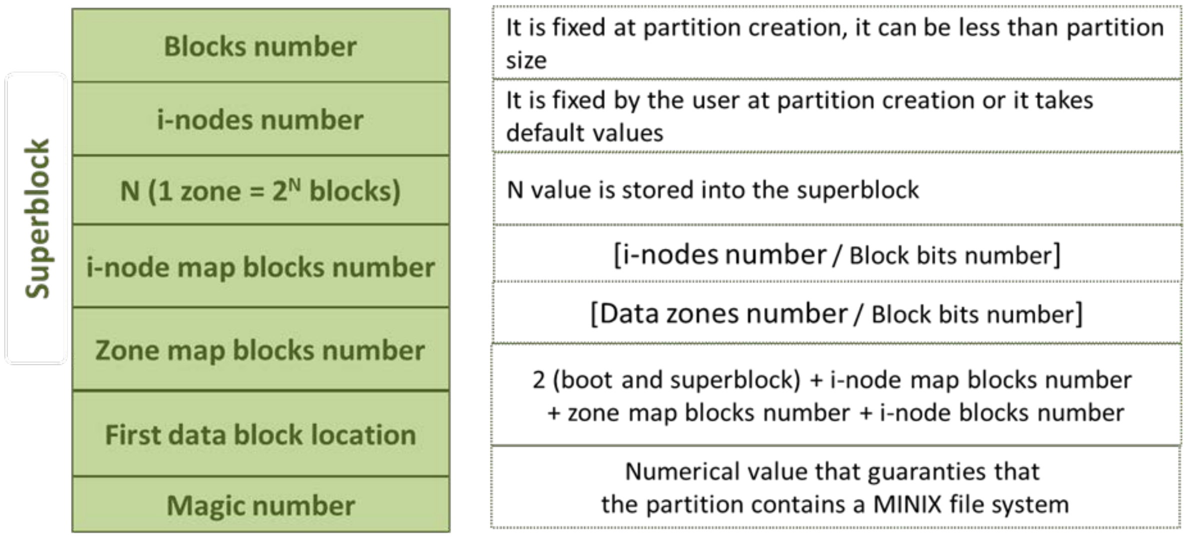
- Directories entries are 16 bytes and contain the number of i-node and file name.



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.



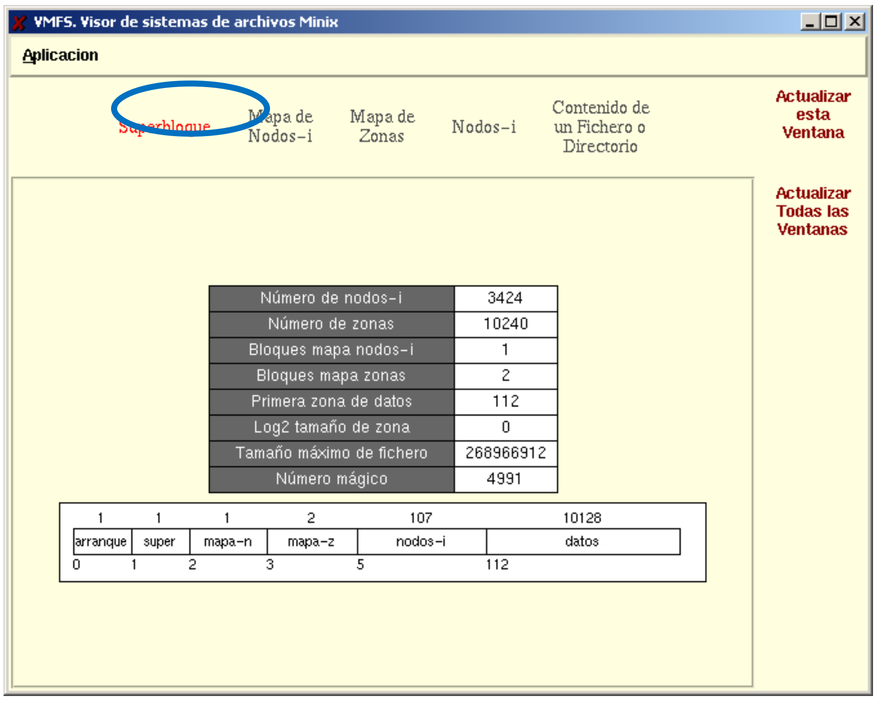
The superblock contains the description of the structure of the partition like its size and location of its elements.



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-nodes number / Block bits number]
Zone map blocks number	[Data zones number / Block bits number]
First data block location	2 (boot and superblock) + i-node map blocks number + zone map blocks number + i-node blocks number
Magic number	Numerical value that guaranties that the partition contains a MINIX file system

6.2 vmfs tool handling

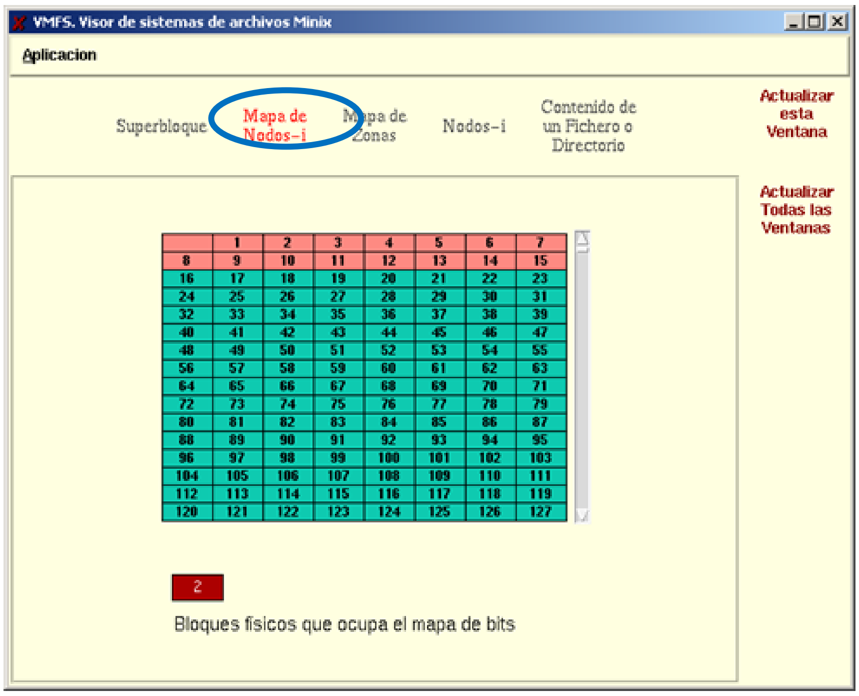
Note. This corresponds to the former application version made with Tcl/Tk. The new version is made on Java but it has basically the same funcionality.
The tool vmfs (viewer of minix file system) was implemented with pedagogical purposes only. vmfs reads the blocks of a device containing a file system 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:


VMFS. Visor de sistemas de archivos Minix
Aplicacion
Superbloque
Mapa de Nodos-i
Mapa de Zonas
Nodos-i
Contenido de un Fichero o Directorio
Actualizar esta Ventana
Actualizar Todas las Ventanas

Número de nodos-i	3424
Número de zonas	10240
Bloques mapa nodos-i	1
Bloques mapa zonas	2
Primera zona de datos	112
Log2 tamaño de zona	0
Tamaño máximo de fichero	268966912
Número mágico	4991

arranque	super	mapa-n	mapa-z	nodos-i	datos
1	1	1	2	107	10128
0	1	2	3	5	112

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

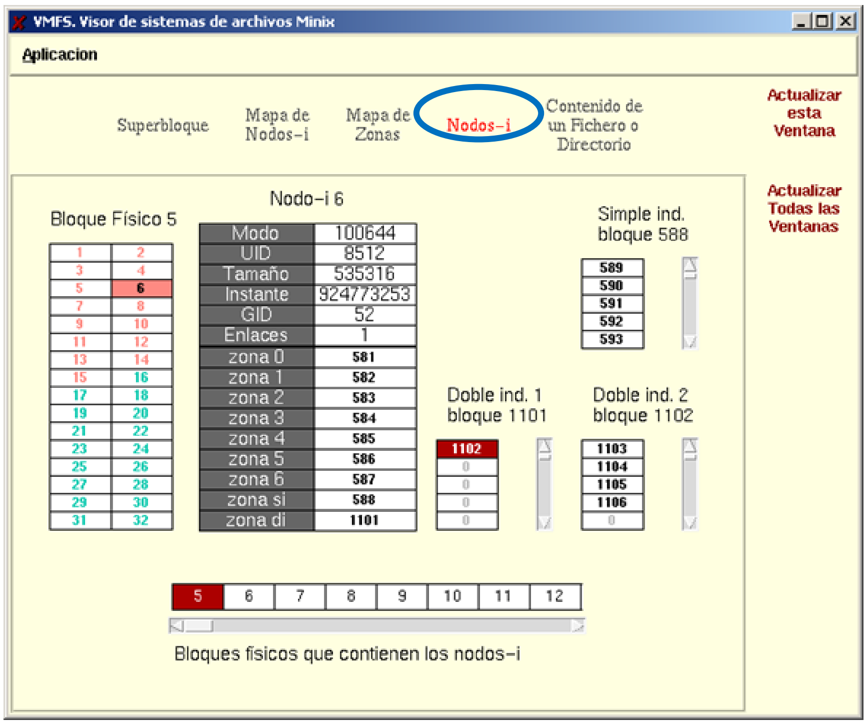

VMFS. Visor de sistemas de archivos Minix
Aplicacion
Superbloque
Mapa de Nodos-i
Mapa de Zonas
Nodos-i
Contenido de un Fichero o Directorio
Actualizar esta Ventana
Actualizar Todas las Ventanas

	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103
104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127

2

Bloques físicos que ocupa el mapa de bits

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.


VMFS. Visor de sistemas de archivos Minix
Aplicacion
Superbloque
Mapa de Nodos-i
Mapa de Zonas
Nodos-i
Contenido de un Fichero o Directorio
Actualizar esta Ventana
Actualizar Todas las Ventanas

Bloque Físico 5

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32

Nodo-i 6

Modo	100644
UID	8512
Tamaño	535316
Instante	924773253
GID	52
Enlaces	1
zona 0	581
zona 1	582
zona 2	583
zona 3	584
zona 4	585
zona 5	586
zona 6	587
zona si	588
zona di	1101

Simple ind. bloque 588

589
590
591
592
593

Doble ind. 1 bloque 1101

1102
0
0
0
0

Doble ind. 2 bloque 1102

1103
1104
1105
1106
0

5	6	7	8	9	10	11	12

Bloques físicos que contienen los nodos-i

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.