

Fundamentos de els Sistemes Operatius (FSO)

Departament de Informàtica de Sistemes i Computadores (DISCA)

Universitat Politècnica de València

Bloc Temàtic 3: Sistema d'Arxius i E/S

Seminari Unitat Temàtica 8

SUT08:

Sistema d'arxius Minix

fSO

DISCA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

- **Objectius**

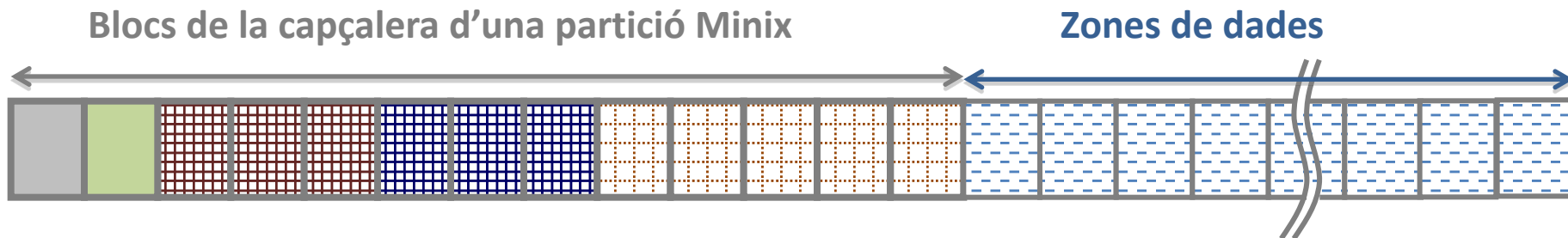
- Descriure l'**estructura d'una partició en Minix**
- Estudiar el **node-i** com a element del sistema operatiu per a mantindre la informació de l'arxiu
- Compendre el concept de **mapa de bits** per a la gestió d'estructures lliures/ocupades
- Ser capaç de **localitzar un arxiu** concret en una estructura de directoris a partir de la seua ruta absoluta

- **Bibliografía**

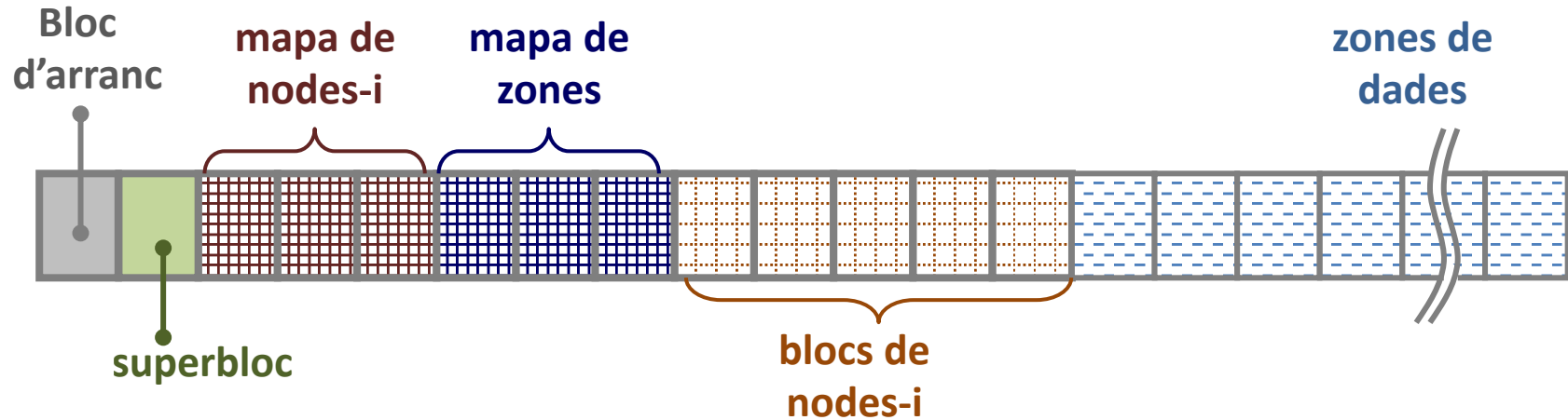
- “Sistemas operativos: Diseño e implementación”, Andrew S. Tanenbaum, Prentice Hall

- **Estructura d'una partició**
- Estructura d'un node-i
- Entrades de directori
- Grandàries estàndard
- Exercicis

- Una **partició Minix** es construeix sobre un entramat de blocs de grandària fixa (p.ex. 1KByte)
 - L'estructura de la partició consta de:
 - La **capçalera** formada per grups de blocs destinats a emmagatzemar les estructures de dades del sistema de fitxers
 - **Zones de dades**, formada per blocs destinats a emmagatzemar la informació pròpia de l'arxiu

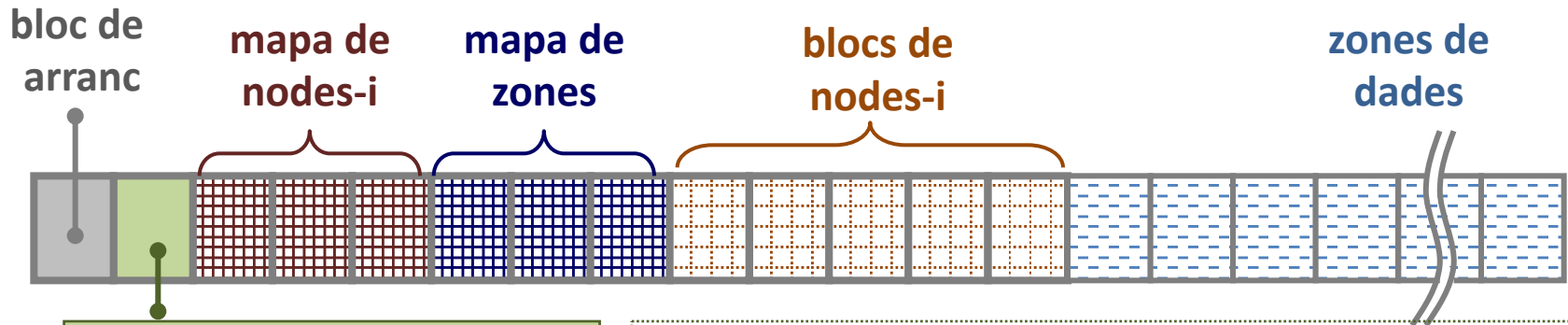


• Partició Minix blocs de la capçalera



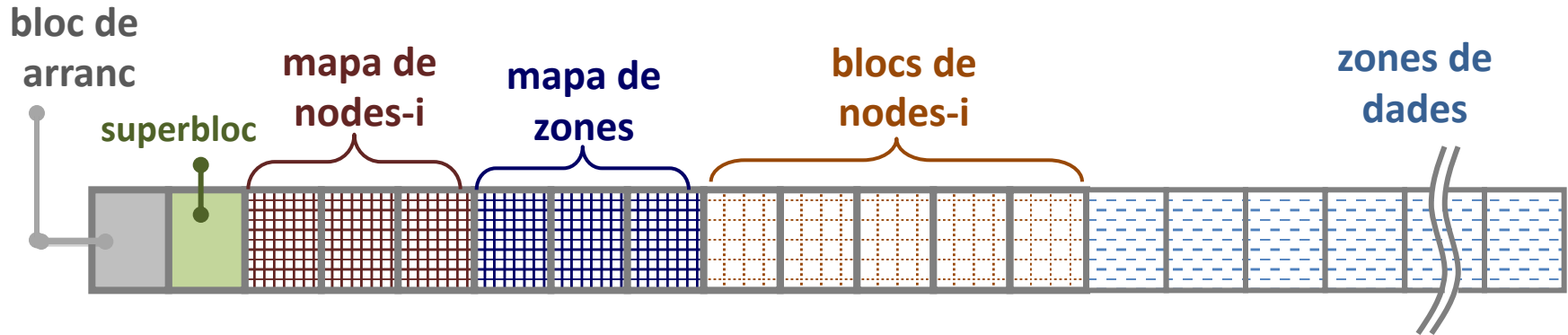
- **Bloc de arranc:** conté el programa d'arranc que carrega el sistema operatiu i li transfereix el control
- **Superbloc:** és una estructura de dades amb la descripció del sistema d'arxius, indica la grandària i ubicació de cada element
- **Mapa bits nodes-i:** vector de bits per a gestionar nodes-i lliures i ocupats. Conté un bit per a cada node-i
- **Mapa bits zones:** vector de bits per a gestionar de zones lliures i ocupades. Conté un bit per cada zona
- **Blocs de nodes-i:** contenen les estructures de dades nodes-i. El nombre de nodes-i depèn del grandària de la partició. El node-i 0 no s'utilitza

• Superbloc de Minix



00	Nombre de nodes-i	El fixa l'usuari en crear la partició o pren valors per defecte
02	Nombre de zones de dades	Es fixa en crear-se la partició. Pot ser menor que la grandària de la partició
04	Nombre de blocs per a el mapa de nodes-i	$[\text{Nombre de nodes-i} / \text{Nombre de bits per bloc}]$
06	Nombre de blocs per a el mapa de zones	$[\text{Nre. de zones de dades} / \text{Nre. de bits per bloc}]$
08	Primer bloc de dades	$2 (\text{arranc i superbloc}) + \text{Nre. de blocs mapa de nodes-i} + \text{Nre. de blocs mapa de blocs} + \text{Nre. de blocs nodes-i}$
0A	$N (1 \text{ zona} = 2^N * 1024)$	$1 \text{ zona} = 2^N \text{ blocs.}$ El valor N s'emmagatzema en el superbloc
0C	Talla màxima d'arxiu	Grandària màxima d'arxiu en bytes
10	Número màgic	Valor numèric que garanteix que aquesta partició conté un sistema d'arxius MINIX

- Mapa de bits/vector de bits



mapa de nodes-i

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Cada **node-i** està **representat** per un **bit** que contindrà 0 si ja ha sigut assignat o 1 si està lliure per a poder-lo assignar a un arxiu

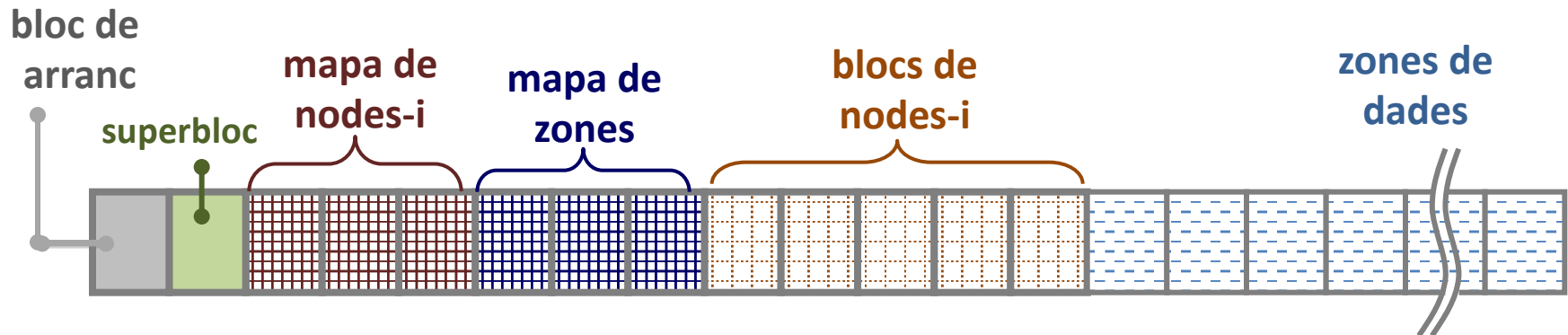
mapa de zones

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	0
1	0	1	1	1	0	0	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Cada **zona de dades** està **representat** per un **bit** que contindrà 0 si ja ha sigut assignat o 1 si està lliure per a ser assignat a un arxiu

bit a 0 assignat
bit a 1 lliure

- **Zona de dades**



- **Zona de dades:** blocs per a emmagatzemar la informació d'arxius regulars, entrades de directori i referències a altres blocs
 - Per a adreçar particions de molta grandària, Minix permet agrupar els blocs de dades en zones
 - La zona es la unitat d'assignació a arxiu
 - **1 zona = 2^N blocs** → per defecte $1 \text{ zona} = 2^0 \text{ blocs} = 1 \text{ bloc}$
 - El primer bloc de dades (valor emmagatzemat en el superbloc) s'ajusta perquè coincidisca amb el començament d'una zona

- Estructura d'una partició
- **Estructura d'un node-i**
- Entrades de directori
- Grandàries estàndard
- Exercicis

Fundamentos de els Sistemes Operatius ETSINF-UPV

- Nombre d'entrades de directori amb aquest node-i



- Anàlisi d'eficiència
 - **Accés aleatòrio eficient:** El nombre màxim d'accessos a disc està limitat a 4
 - Els punters indirectes només s'utilitzen per a fitxers grans i molt grans (que són pocs)
 - Per als fitxers menuts l'accés és molt eficient
 - **Diseny elegant i fiable:** cada fitxer té la seua estructura de dades separada

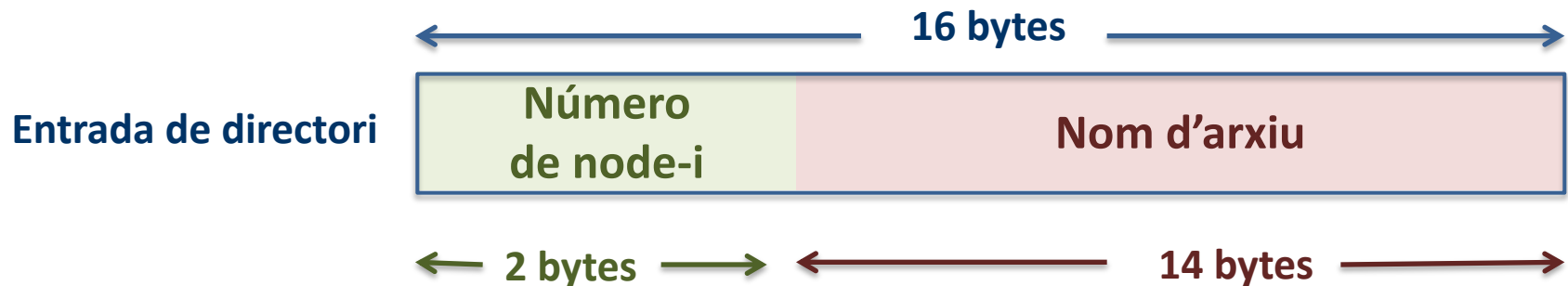
- Estructura d'una partició
- Estructura d'un node-i
- **Entrades de directori**
- Grandàries estàndard
- Exercicis

- **Directoris en Minix**

- Estructura de directoris com a graf acíclic dirigit (DAG)
- Els directoris són arxius els bytes dels quals s'interpreten com a registres → entrades de directoris o enllaços

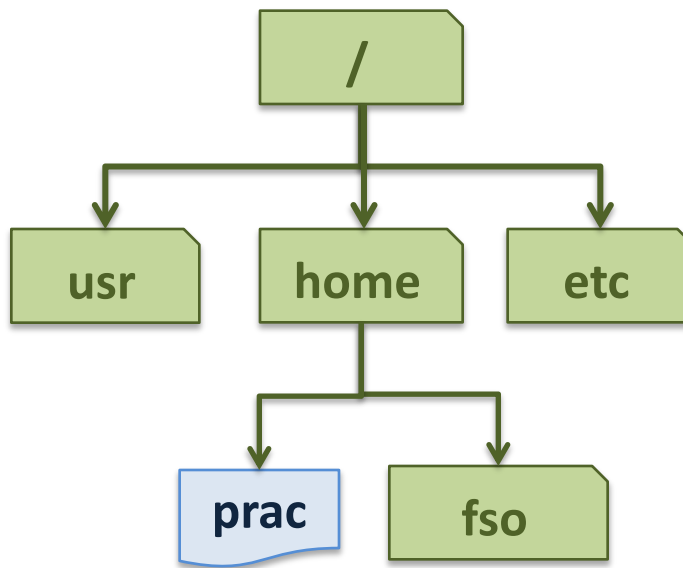
- **Entrada de directori Minix**

- L'entrada de directori també es denomina *enllaç*
- Amb una grandària de 16 bytes
 - 2 bytes per al número de node-i
 - 14 bytes per al nom de l'arxiu



- **Entrades de directori**

- Quan es crea un directori, es creen **les entrades ‘.’ i ‘..’** automàticament.
- El **node-i 1** descriu el directori arrel
- Quan s’esborra una entrada es marca amb el node-i 0



1	.
1	..
3	usr
4	home
5	etc

4	.
1	..
35	prac
84	fso

84	.
4	..

3	.
1	..

5	.
1	..

- Estructura d'una partició
- Estructura d'un node-i
- Entrades de directoris
- **Grandàries estàndard**
- Exercicis

- Grandàries per defecte per als diferents elements de Minix
 - 1 zona = 2^0 blocs = 1024 bytes
 - 1 punter a zona o bloc = 2 bytes = 16 bits
 - 1 entrada de directori = 16 bytes
 - 1 node-i = 32 bytes

- Estructura d'una partició
- Estructura d'un node-i
- Entrades de directoris
- Grandàries estàndard
- **Exercicis**

• Exercici 1: Grandària màxima d'un arxiu

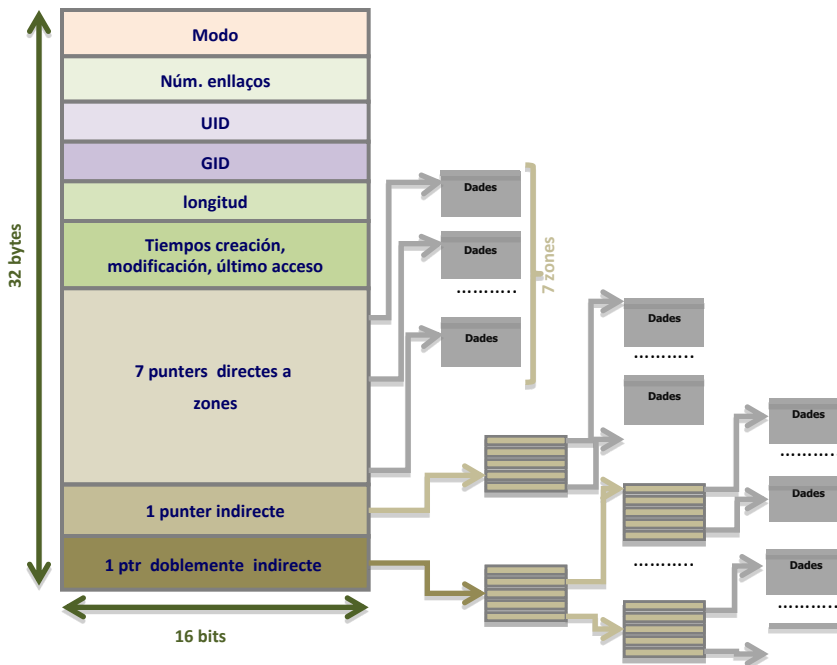
Determineu la grandària màxima teòrica (sense tindre en compte la grandària real del dispositiu sobre el què ha de residir l'arxiu) d'un arxiu Minix. Especifiqueu els blocs adreçats amb cada tipus de punter. Els paràmetres de Minix a considerar són:

Punters a zones de dades de 16 bits

Grandària de bloc 1K

1 zona = 1 bloc

Estructura de node-i: 7 punters directes, 1 indirecte i 1 doble indirecte



Solució:

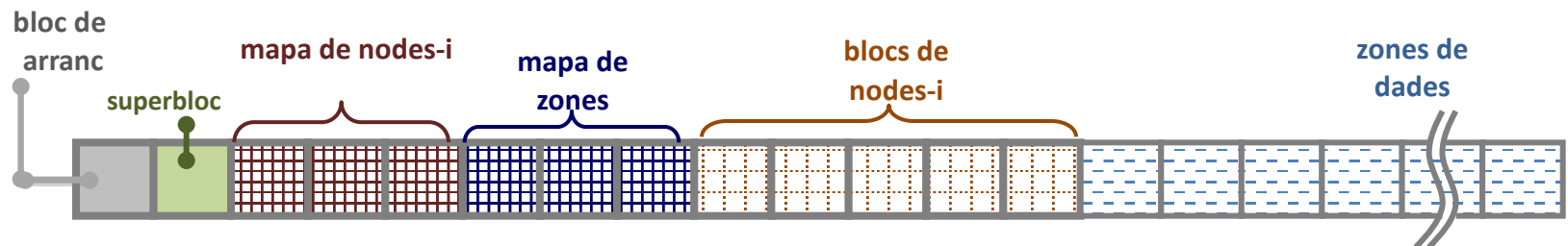
$$7 \text{ Kb} + 512 \text{ Kb} + 262.144 \text{ Kb} = 262.633 \text{ Kb (aprox. 256.5 Mb)}$$

• Exercici 2: Càlcul de l'estructura d'una partició

Donat un disc de 20 MBytes en MINIX amb els paràmetres següents:

- Punters a zones de dades de 16 bits
- Grandària de bloc 1K (1 zona = 1 bloc)
- Grandària de node-i de 32 bytes
- Nombre màxim de nodes-i: 512

- Especifiqueu clarament totes les estructures de dades que formen el sistema d'arxius i els blocs que n'ocupa cadascuna
- En cas de resultar danyada l'estructura del mapa de bits de zones, penseu la forma de reconstruir-la amb la informació de la que es disposa en la resta d'estructures del sistema d'arxius (suposeu que la resta de les estructures estan correctes).



• Exercici 3: Recerca d'un arxiu en un directori

Siga un sistema d'arxius Minix, creat amb les grandàries estàndard, amb les següents entrades a directori:

1	.
1	..
3	usr
4	home
5	etc

4	.
1	..
35	prac
84	fso

3	.
1	..
60	ut12
61	sut12

84	.
4	..
90	map
91	listado

5	.
1	..

90	.
84	..

- Dibuixeu l'arbre de directoris i arxius que correspon a aquest sistema
- Indiqueu, de forma justificada, **quins números de nodes-i i a quants blocs** cal accedir, si es vol fer la lectura dels 128 primers bytes de l'arxiu **/home/fso/listado**
- Quina informació hauriem d'aconseguir al llegir els 32 primers bytes de l'arxiu **/home/fso/map**