Fundamentos de los Sistemas Operativos (FSO)

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

Bloque Temático 3: Sistema de Archivos y E/S
Seminario Unidad Temática 8

SUT08:
Sistema de archivos Minix

fSO

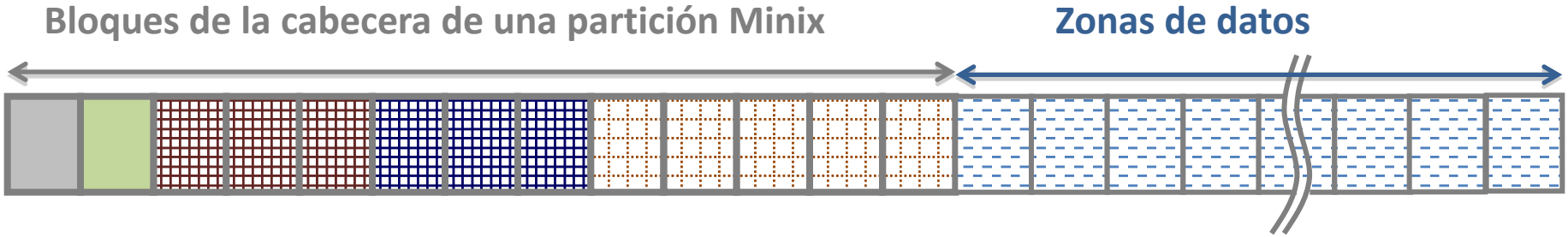
Sistema de archivos Minix

- **Objetivos**
  - Describir la **estructura de una partición en Minix**
  - Estudiar el **nodo-i** como elemento del sistema operativo para mantener la información del archivo
  - Comprender el concepto de **mapa de bits** para la gestión de estructuras libre/ocupado
  - Ser capaz de **localizar un archivo** concreto en una estructura de directorios a partir su ruta absoluta
- **Bibliografía**
  - "Sistemas operativos: Diseño e implementación", Andrew S. Tanenbaum, Prentice Hall

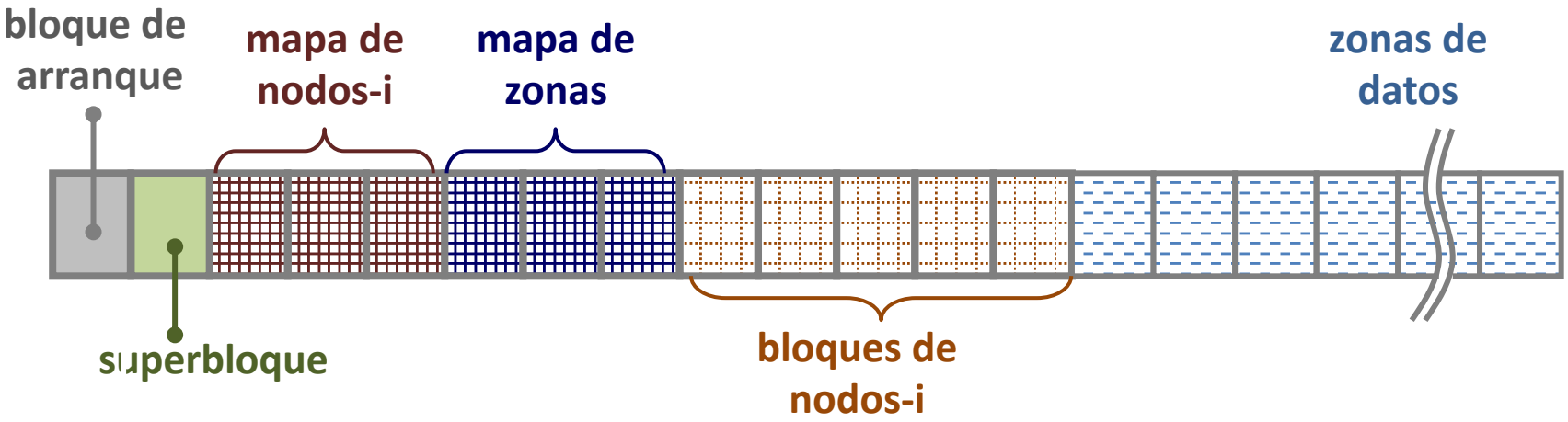
- **Estructura de una partición**
- Estructura de un nodo-i
- Entradas de directorio
- Tamaños estándar
- Ejercicios

Estructura de una partición

- **Partición Minix** se construye sobre un entramado de bloques de tamaño fijo (p.ej. 1KByte)
  - La estructura de la partición consta de:
    - La **cabecera** formada por grupos de bloques destinados a almacenar las estructuras de datos del sistema de ficheros
    - **Zonas de datos**, formada por bloques destinados a almacenar la información propia del archivo

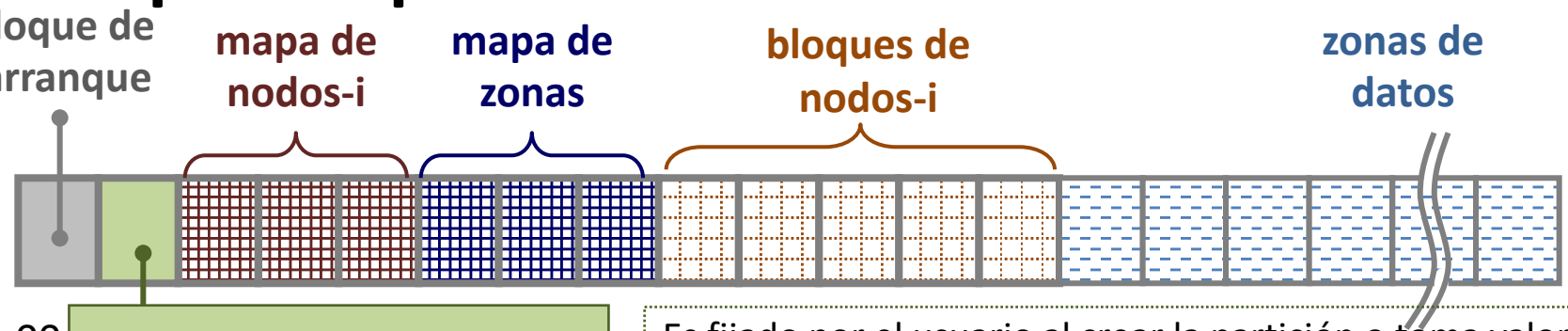


- **Partición Minix bloques de la cabecera**



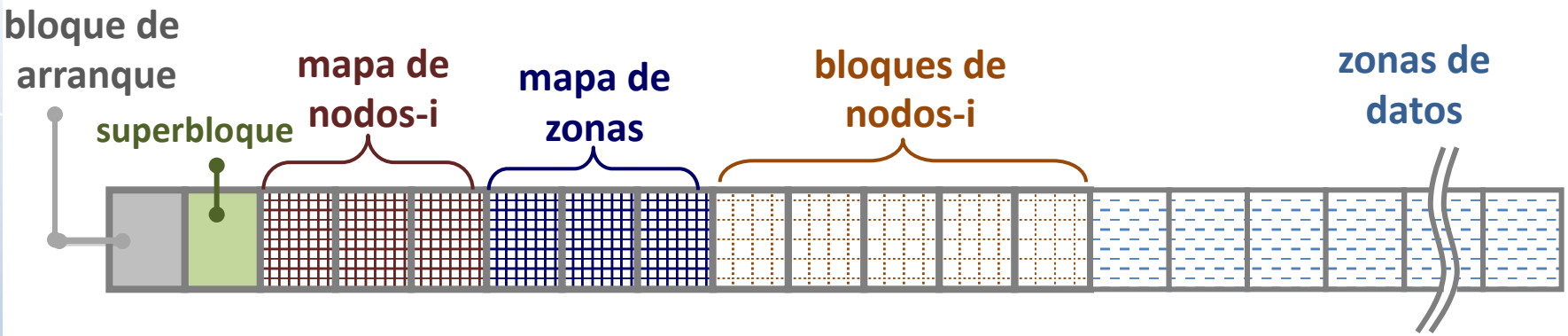
- **Bloque de arranque:** contiene el programa de arranque que carga el sistema operativo y le transfiere control
- **Superbloque:** es una estructura de datos con la descripción del sistema de archivos, indica tamaño y ubicación de cada elemento
- **Mapa bits nodos-i**: vector de bits para gestionar nodos-i libres y ocupados. Contiene un bit por cada nodo-i
- **Mapa bits zonas**: vector de bits para gestionar de zonas libres y ocupadas. Contiene un bit por cada zona
- **Bloques de nodos-i:** contienen las estructuras de datos nodos-i. El número de nodos-i depende del tamaño de la partición. El nodo-i 0 no se utiliza

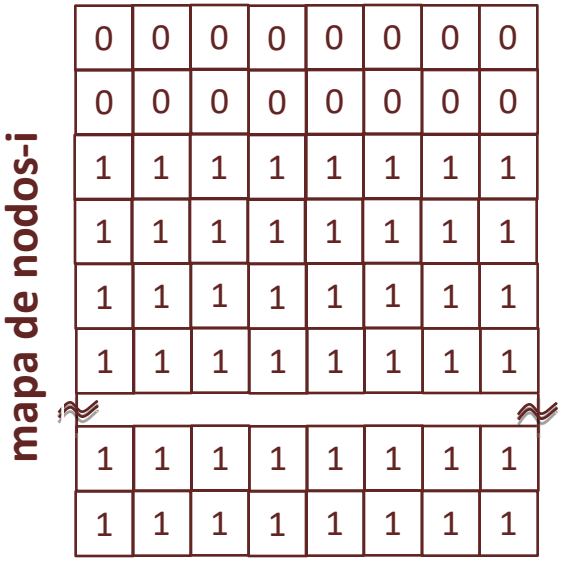
- **Superbloque de Minix**

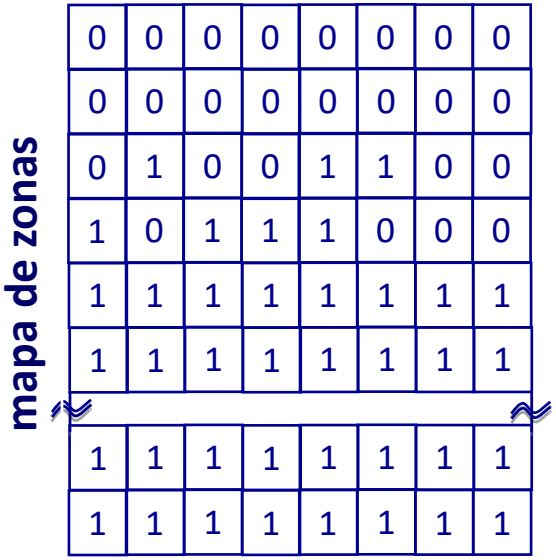


	Superbloque	
00	Número de nodos-i	Es fijado por el usuario al crear la partición o toma valores por defecto
02	Número de zonas de datos	Es fijado al crear la partición, puede ser menor que el tamaño de la partición
04	Número de bloques para el mapa de nodos-i	[Número de nodos-i / Número de bits por bloque]
06	Número de bloques para el mapa de zonas	[Número de zonas de datos / Número de bits por bloque]
08	Nº de bloque de la primera zona de datos	2 (Bloque de arranque y superbloque) + Número de bloques para el mapa de nodos-i + Número de bloques para el mapa de bloques + Número de bloques de nodos-i
0A	N (1 zona= 2^N * 1024)	1 zona = 2^N bloques El valor N se almacena en el superbloque
0C	Tamaño máximo de archivo	Tamaño máximo de archivo en bytes
10	Número mágico	Valor numérico que garantiza que esta partición contiene un sistema de archivos MINIX

- **Mapa de bits/vector de bits**





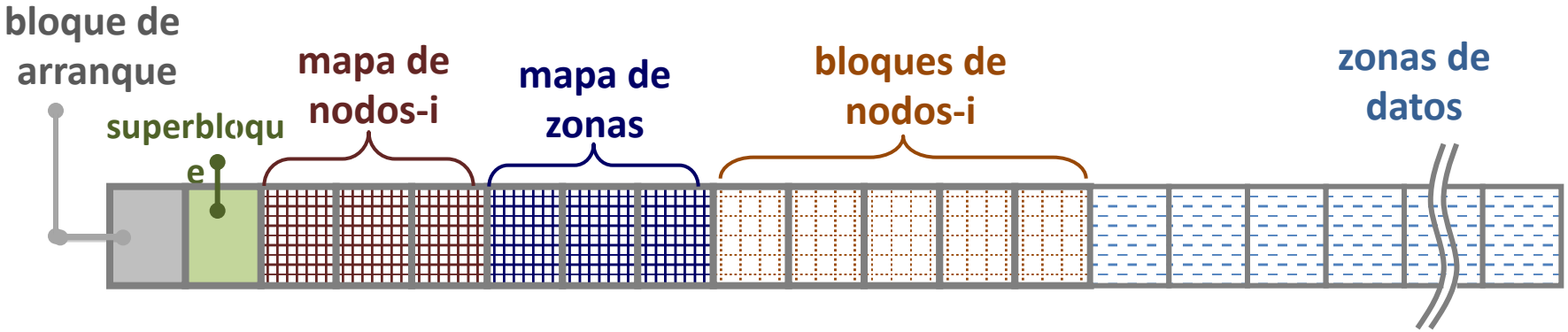


bit a 0 asignado
bit a 1 libre

Cada **nodo-i** esta **representado** por **un bit** que contendrá 0 ó 1 si ya ha sido asignado o esta libre para ser asignado a un archivo

Cada **zona de datos** esta **representado** por **un bit** que contendrá 0 ó 1 si ya ha sido asignado o esta libre para ser asignado a un archivo

- **Zona de datos**



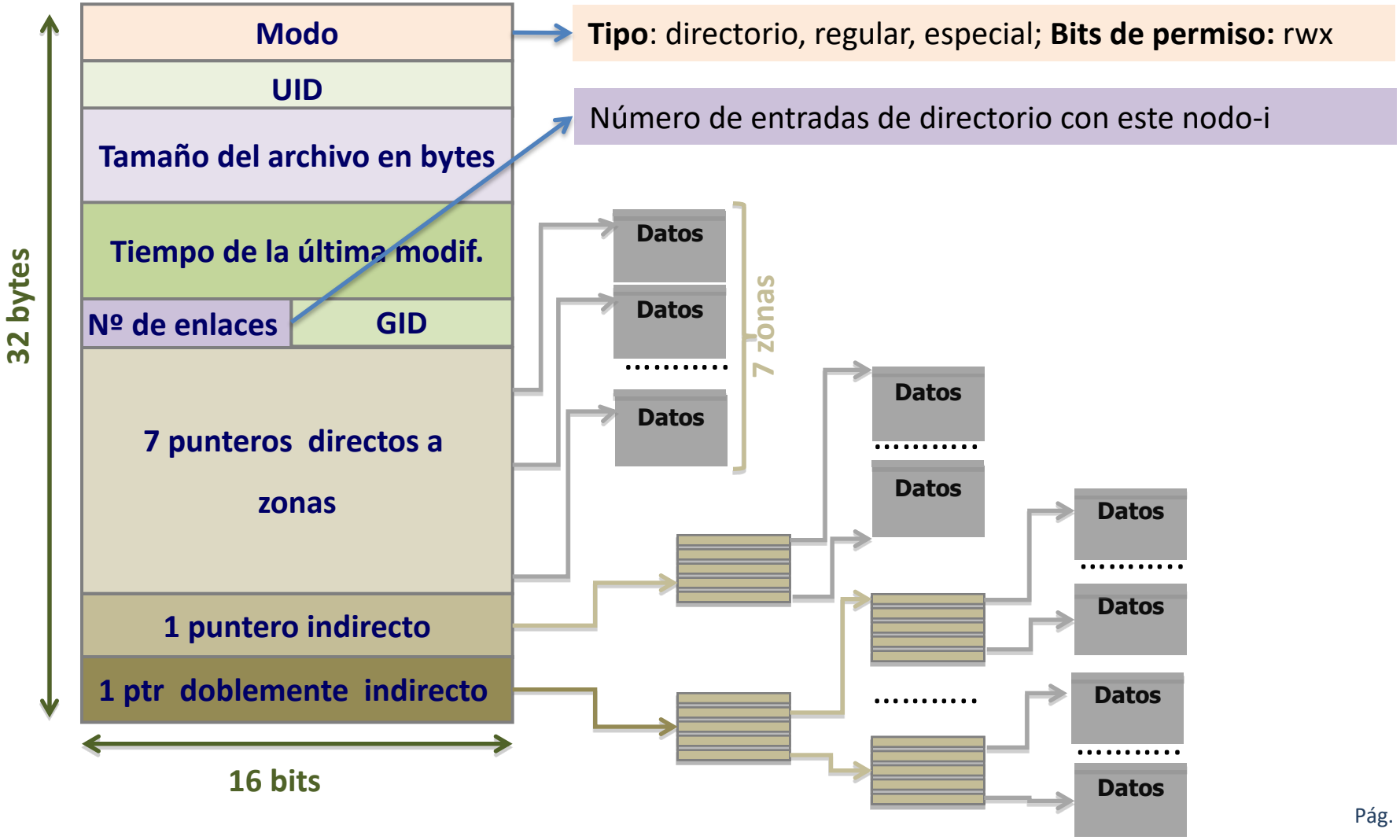
- **Zona de datos:** bloques para almacenar la información de archivos regulares, entradas de directorios y referencias a otros bloques
  - Para direccionar particiones de gran tamaño, Minix permite agrupar los bloques de datos en zonas
  - La zona es la unidad de asignación a archivo
    - **1 zona = 2^N bloques** → por defecto 1 zona = 2^0 bloques = 1bloque
    - El primer bloque de datos (valor almacenado en el superbloque) se ajusta para que coincida con el comienzo de una zona

Contenido

- Estructura de una partición
- **Estructura de un nodo-i**
- Entradas de directorio
- Tamaños estándar
- Ejercicios

Estructura de un nodo-i

- **Nodo-i de Minix**
  - Estructura de datos con todos los atributos del archivo excepto el nombre
    - Cada archivo en Minix se le asigna un nodo-i
    - Asignación indexada con punteros a bloques directos, indirectos y doblemente indirectos



- **Nodo-i Minix**
  - Análisis de eficiencia
    - **Acceso aleatorio eficiente:** El número máximo de accesos a disco está limitado a 4
      - Los punteros indirectos sólo se utilizan para ficheros grandes y muy grandes (que son escasos)
      - Para los ficheros pequeños el acceso es muy eficiente
    - **Diseño elegante y fiable**: cada fichero tiene su estructura de datos separada

Contenido

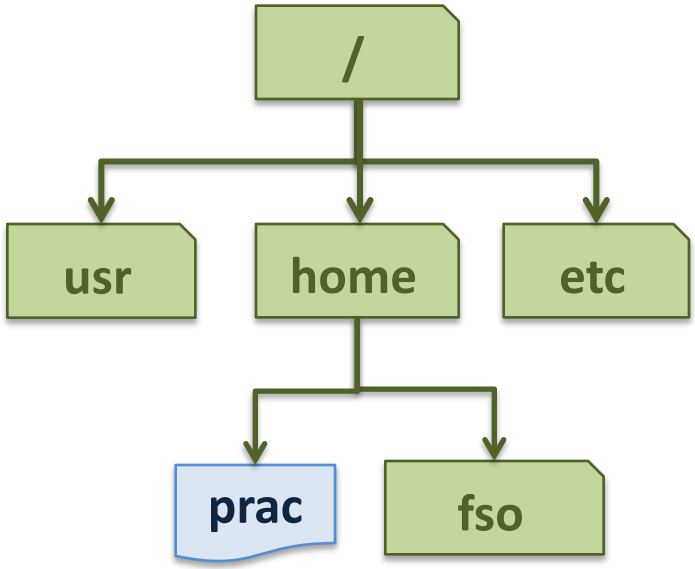
- Estructura de una partición
- Estructura de un nodo-i
- **Entradas de directorio**
- Tamaños estándar
- Ejercicios

Entradas de directorios

- **Directorios en Minix**
  - Estructura de directorios como grafo acíclico dirigido (DAG)
  - Los directorios son archivos cuyos bytes se interpreta como registros → entradas de directorios o enlaces
- **Entrada de directorio Minix**
  - La entrada de directorio también se denominada enlace
  - Con un tamaño de 16 bytes
    - 2 bytes para el número de nodo-i
    - 14 bytes para el nombre del archivo

Entradas de directorios

- **Entradas de directorio**
  - Cuando se crea un directorio, se crean **las entradas '.' y '..'** automáticamente.
  - El **nodo-i 1 describe al directorio raíz**
  - Cuando se borra una entrada se marca con el nodo-i 0



1	.
1	..
3	usr
4	home
5	etc

4	.
1	..
35	prac
84	fso

84	.
4	..

3	.
1	..

5	.
1	..

Contenido

- Estructura de una partición
- Estructura de un nodo-i
- Entradas de directorio
- **Tamaños estándar**
- Ejercicios

Tamaños estándar

- Tamaños por defecto para los diferentes elementos de Minix
  - 1 Zona =2^0 bloques = 1024 bytes
  - 1 Puntero a zona o bloque = 2 bytes = 16 bits
  - 1 Entrada de directorio = 16 bytes
  - 1 Nodo-i = 32 bytes

Contenido

- Estructura de una partición
- Estructura de un nodo-i
- Entradas de directorio
- Tamaños estándar
- **Ejercicios**

- **Ejercicio 1: Tamaño máximo de un archivo**

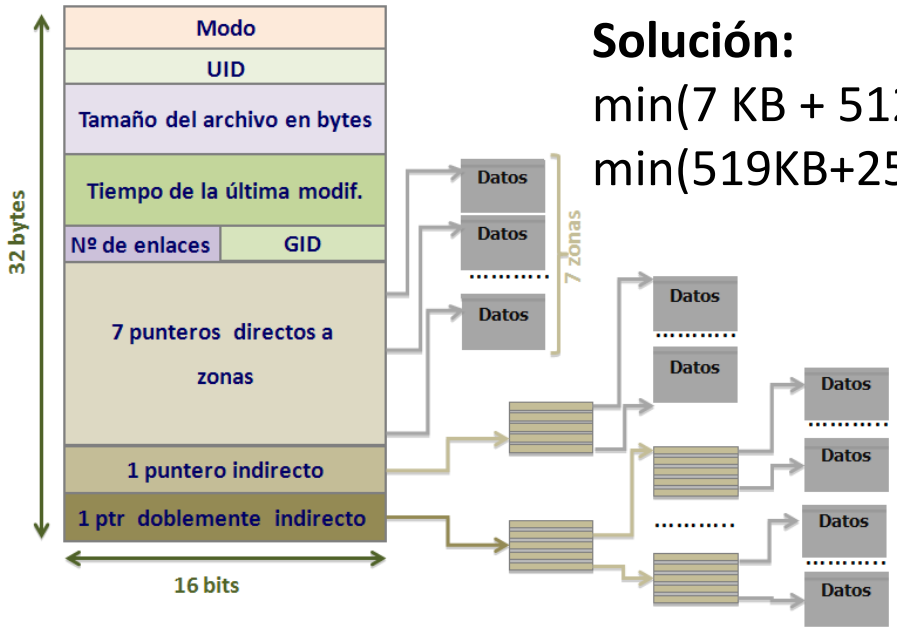
  Determine el máximo tamaño teórico (no tenga en cuenta el tamaño real del dispositivo sobre el que deba residir el archivo) de un archivo Minix. Especifique los bloques direccionados con cada tipo de puntero. Los parámetros de Minix ha considerar son:

  Punteros a zonas de datos de 16 bits

  Tamaño de bloque 1K

  1 zona = 1 bloque

  Estructura de nodo-i: 7 punteros directos,1 indirecto y 1 doble indirecto



Solución:

min(7 KB + 512 KB + 512*512*1KB, 64K*1KB) =

min(519KB+256K*1KB, 64K*1KB) = 64MB

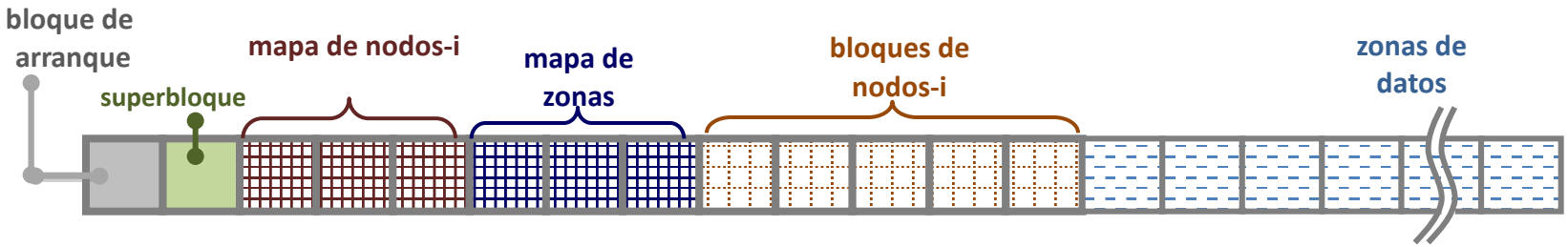
- **Ejercicio 2: Cálculo de la estructura de una partición**

  Dado un disco de 20 Mb en MINIX con los siguientes parámetros:

  - Punteros a zonas de datos de 16 bits
  - Tamaño de bloque 1K (1 zona = 1 bloque)
  - Tamaño de nodo-i de 32 bytes
  - Número máximo de nodos-i: 512

  a) Especifique claramente todas las estructuras de datos que forman el sistema de archivos y los bloques que ocupa cada una

  b) En caso de resultar dañada la estructura del mapa de bits de zonas, piense la forma de reconstruirla con la información de la que se dispone en el resto de estructuras del sistema de archivos (suponga que el resto de las estructuras están inalteradas).



- **Ejercicio 3: Búsqueda de un archivo en un directorio**

Sea un sistema de archivos Minix, creado con los tamaños estándar cuyas entradas de directorio actualmente son la siguientes

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	..

a) Dibuje el árbol de directorios y archivos que corresponde a dicho sistema

b) Indique de forma justificada **que números de nodos-i y cuantos bloques** es necesario acceder, si se requiere la lectura de los 128 primeros bytes del archivo **/home/fso/listado**

c) Que información deberíamos conseguir al leer los 32 primeros bytes del archivo **/home/fso/map**