file management

File Management

- File management system consists of system utility software that run as privileged applications
  - usually part of the os kernel
  - usually based on block-oriented devices (i.e. disks)

File System Properties

- Structure
- Long-term existence
- Sharable between processes

File Operations

- Create
- Delete
- Open
- Close
- Read
- Write

Terms Used with Files

- Field
  - Basic element of data
  - Contains a single value
  - Characterized by its length and data type
- Record
  - Collection of related fields
  - Treated as a unit
    - Example: employee record

Terms Used with Files

- File
  - Collection of similar records
  - Treated as a single entity
  - Have file names
  - May restrict access
- Database
  - Collection of related data
  - Relationships exist among elements

Objectives for a File Management System

- Meet the data management needs and requirements of the user
- Guarantee that the data in the file are valid
- Optimize performance
- Provide I/O support for a variety of storage device types

Objectives for a File Management System

- Minimize or eliminate the potential for lost or destroyed data
- Provide a standardized set of I/O interface routines
- Provide I/O support for multiple users

file systems need data structures

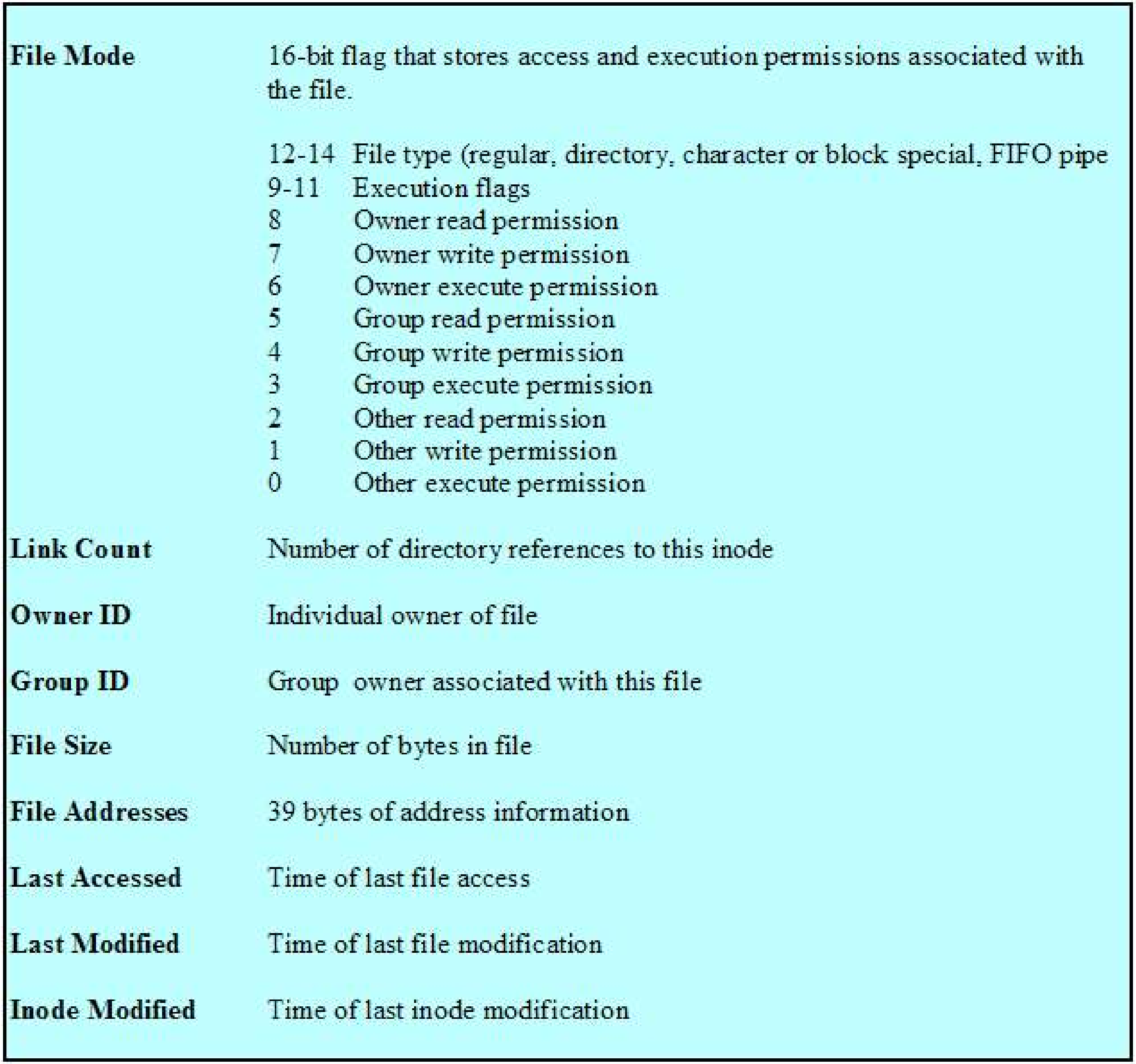
- to keep track of which blocks are allocated to a given file
- to keep track of free blocks
- to keep names, directories, access rights, etc

UNIX File Management

- Types of files
  - Regular, or ordinary
    - a sequence of bytes (no records!)
  - Directory
  - Special (character of block devices)
  - Named pipes (FIFO)
  - Links (hard links)
  - Symbolic links (soft links)

Inodes

- Index node
- Control structure that contains key information for a particular file



File Mode	16-bit flag that stores access and execution permissions associated with the file. 12-14 File type (regular, directory, character or block special, FIFO pipe 9-11 Execution flags 8 Owner read permission 7 Owner write permission 6 Owner execute permission 5 Group read permission 4 Group write permission 3 Group execute permission 2 Other read permission 1 Other write permission 0 Other execute permission
Link Count	Number of directory references to this inode
Owner ID	Individual owner of file
Group ID	Group owner associated with this file
File Size	Number of bytes in file
File Addresses	39 bytes of address information
Last Accessed	Time of last file access
Last Modified	Time of last file modification
Inode Modified	Time of last inode modification

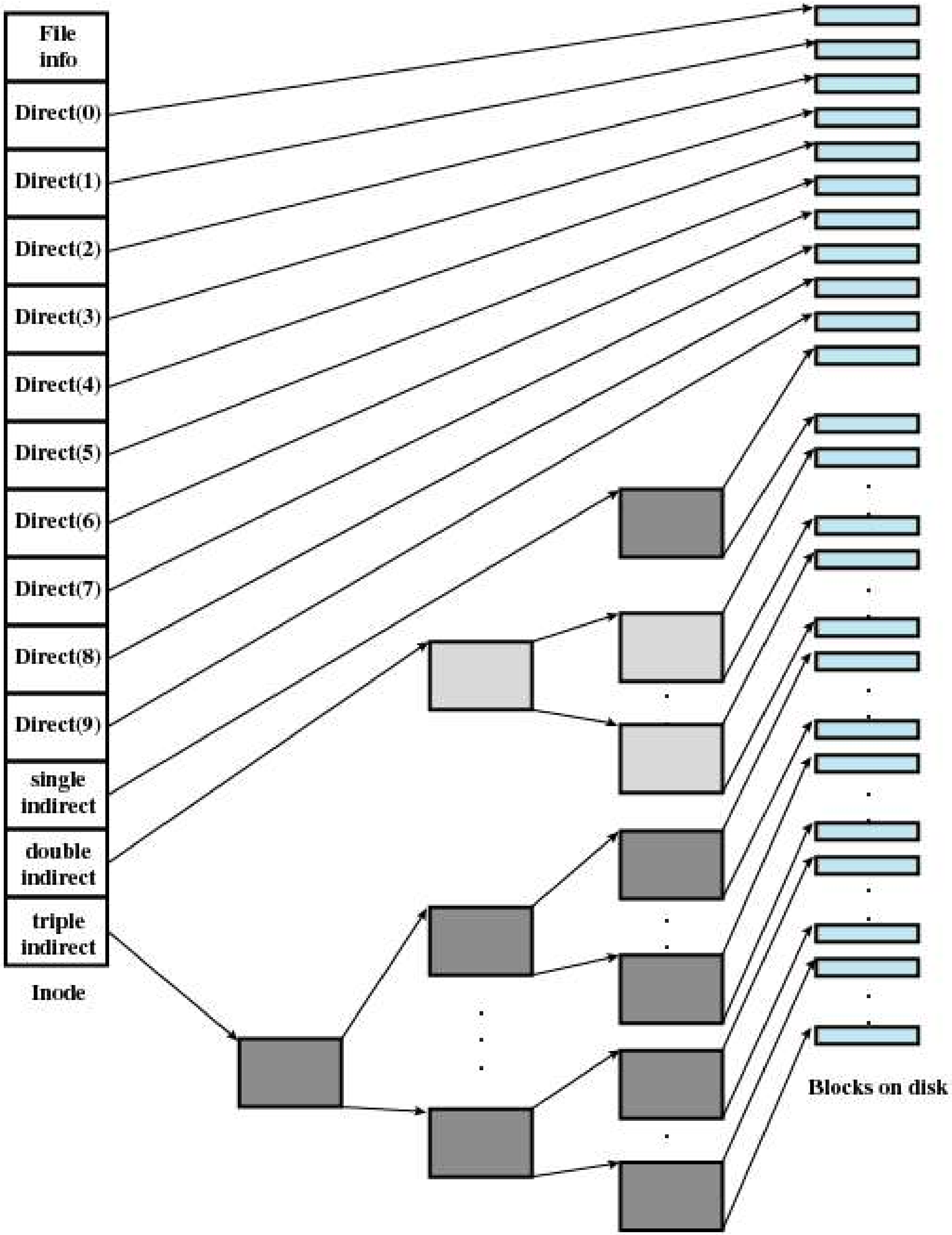


Figure 12.13 Layout of a UNIX File on Disk

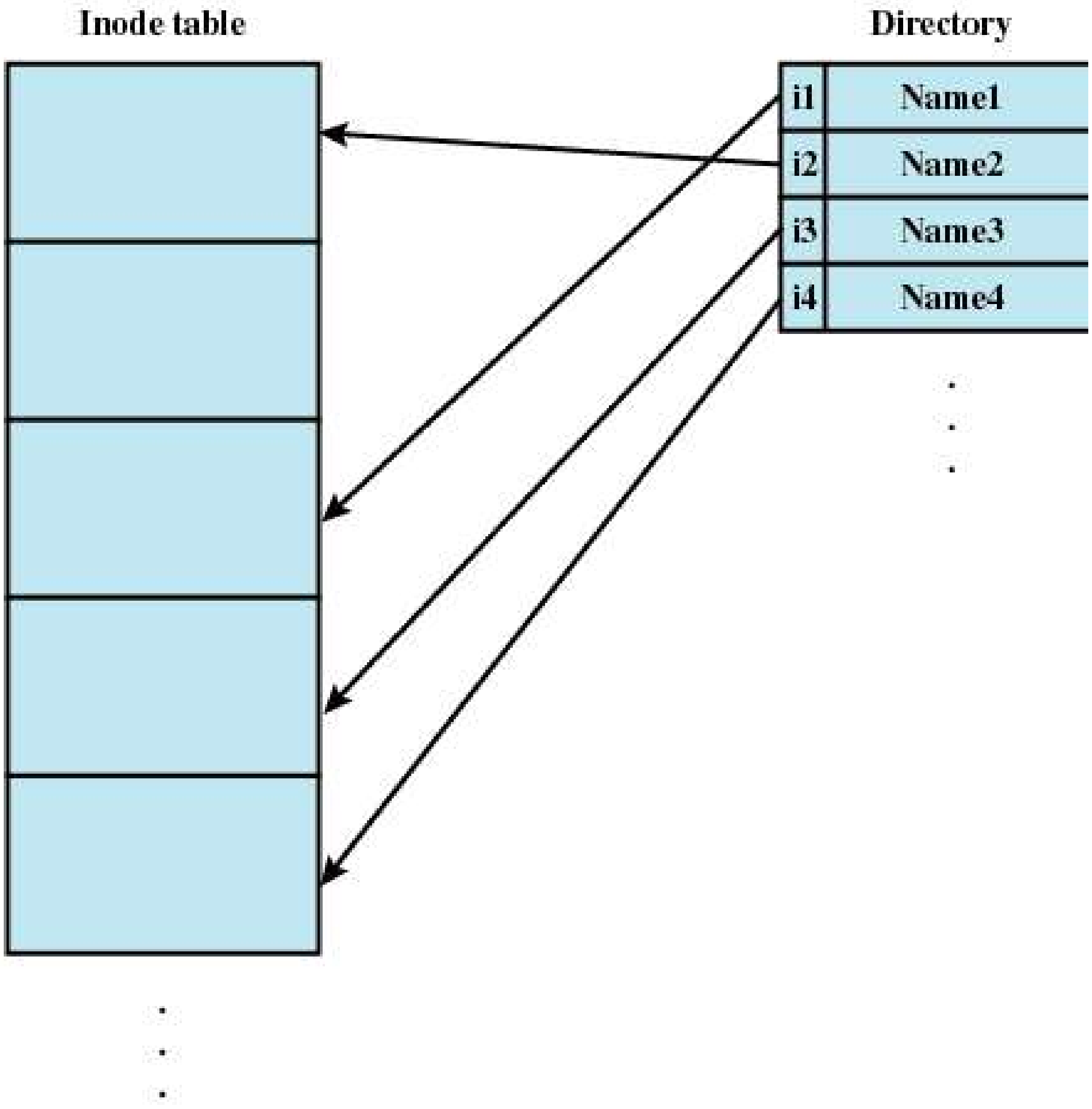


Figure 12.14 UNIX Directories and Inodes

hard and soft links

- hard links file
  - more filenames can point to the same inode
  - each name is said **hard link**
  - an hard link is always valid
  - an inode is deleted if and only if it has no names
    - reference counting
- soft links
  - a soft link is a file that contains a pathname
  - they can be relative or absolute
  - they can be invalid

Linux Virtual File System

- Uniform file system interface to user processes
- Represents any conceivable file system's general feature and behavior
- Assumes files are objects that share basic properties regardless of the target file system

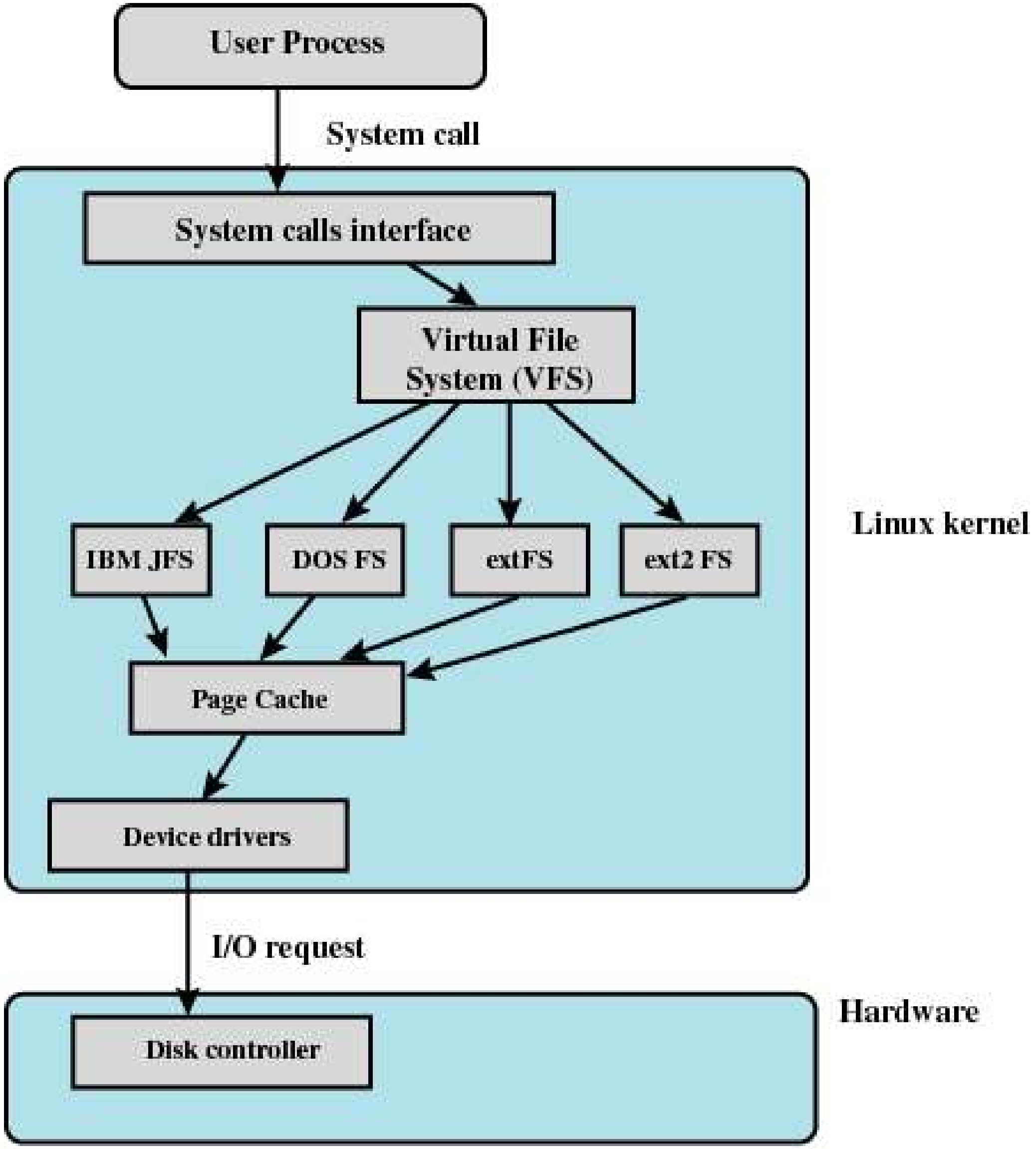
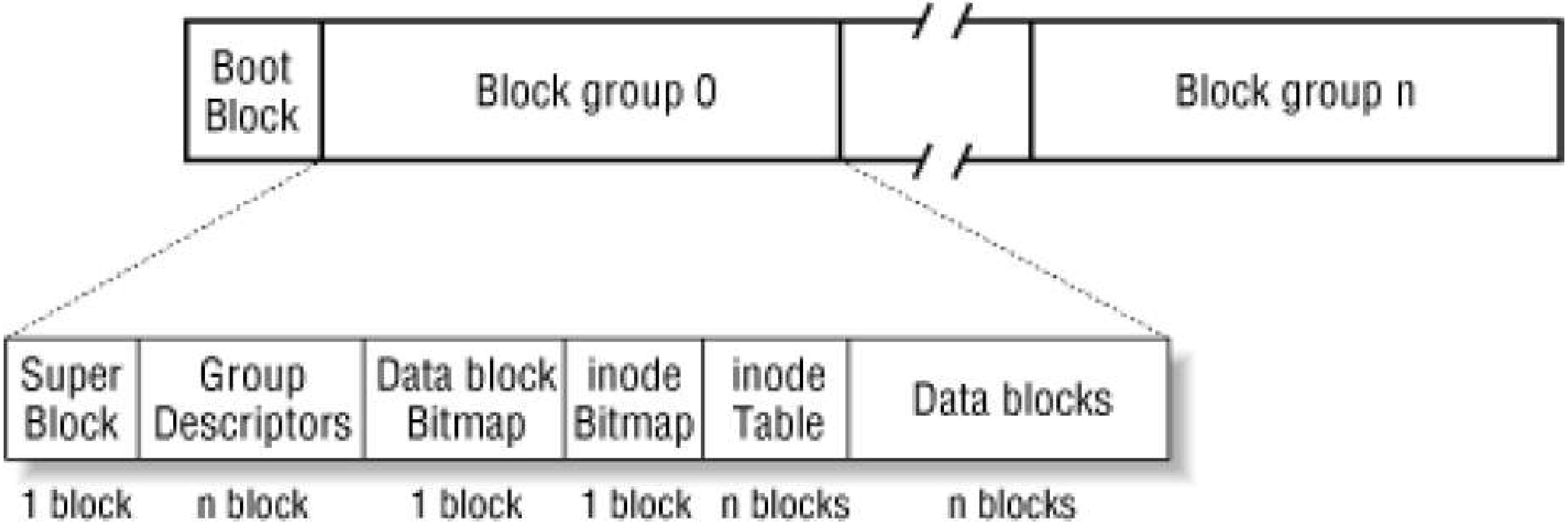


Figure 12.15 Linux Virtual File System Context

Primary Objects in VFS

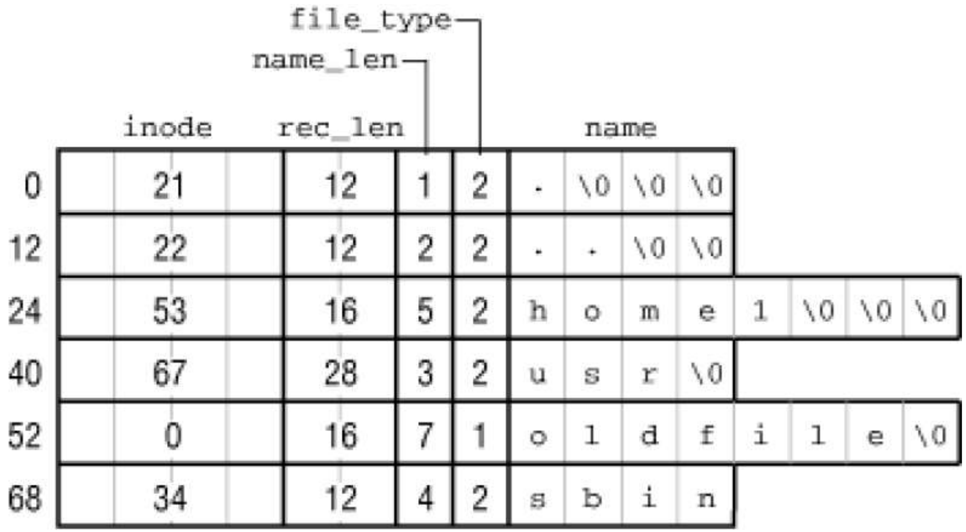
- Superblock object
  - Represents a specific mounted file system
- Inode object
  - Represents a specific file
- Dentry object
  - Represents a specific directory entry
- File object
  - Represents an open file associated with a process

ext2 file system



- for each group
  - all groups are of the same length
  - each group says where all other groups are (redundancy)
  - 2 bitmaps for free/allocated spaces
    - both data blocks and inodes of this group

ext2 file system



	inode	rec_len	name_len	file_type	name
0	21	12	1	2	. \0 \0 \0
12	22	12	2	2	. . \0 \0
24	53	16	5	2	h o m e 1 \0 \0 \0
40	67	28	3	2	u s r \0
52	0	16	7	1	o l d f i l e \0
68	34	12	4	2	s b i n

- directory stored in variable record length format
- stored as a file
- given the inode *i* number where is it stored on the disk?

  group: *i* / inodes_in_one_group

  inode: i % inodes_in_one_group