# The Linux File System

# *Introduction*

- “File System” is a group of files and relevant information regarding them.
- The disk space allotted to a Linux File System is made up of 'blocks', each of which are of 512bytes
- To find out block size on your file system, use the cmchk command
- All the blocks belonging to the file system are logically divided into:
  - **“Boot Block”**
  - **“Super Block”**
  - **“Inode Table”**
  - **“Data Blocks”**

# The Boot Block

- Represents the beginning of the file system
- Contains a program “bootstrap loader”
- This program is executed when we 'boot' the host machine
- All file system contain one (possibly empty) boot block

# The Super Block

- Describes the state of the file system
- State – how large it is, how many maximum files it can accommodate, how many more files can be created, etc.

# The Inode Table

- The information related to all the files (not the contents) is stored in an Inode Table on the disk
- For each file, there is an inode entry in the table
- Each entry is made up of 64 bytes & contains the details for that file
- These Details are:

a)Owner of the file

b)Group to which the owner belongs

c)Type of file

d)File Access Permission

e) Date and Time of last access
f) Date and Time of last modifications
g) Number of links to the file
h) Size of the file
i) Addresses of blocks where the file is physically present

# Data Blocks

- Contains the actual file contents
- An allocated block can belong to only one file in the file system
- This block can't be used for storing any other file's contents unless the file to which it originally belonged is deleted

# NOTE:

Information stored in the Inode Table must change whenever we use any file , or change its permissions, etc  => Would gobble up a lot of precious CPU time

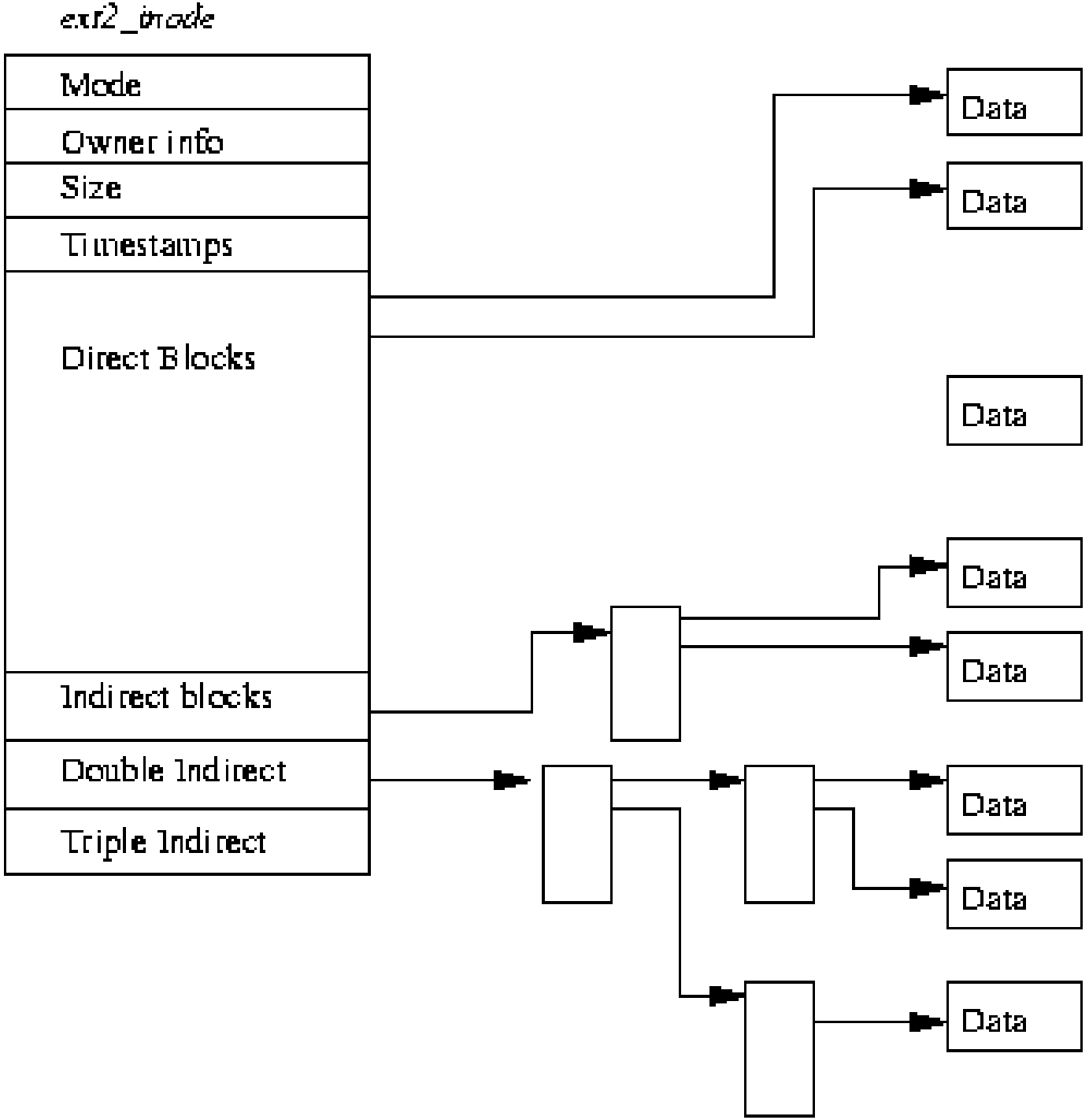
To remedy this, a copy of the Super Block and Inode Table gets loaded into RAM at start-up time

'sync' command synchronises the inode table in memory with the one on disk by simply overwriting the memory copy on to the disk

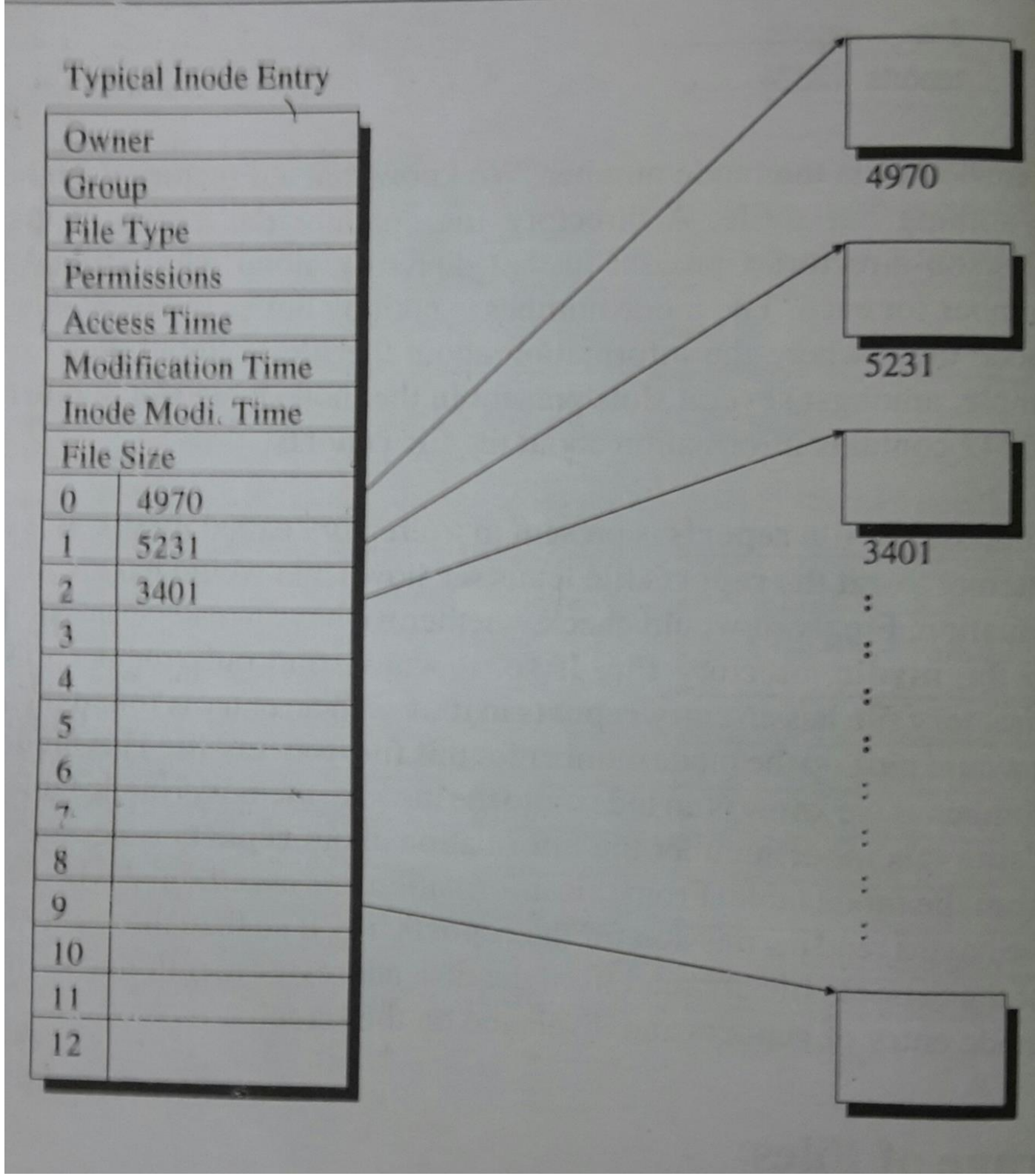
# *How does LINUX access files?*

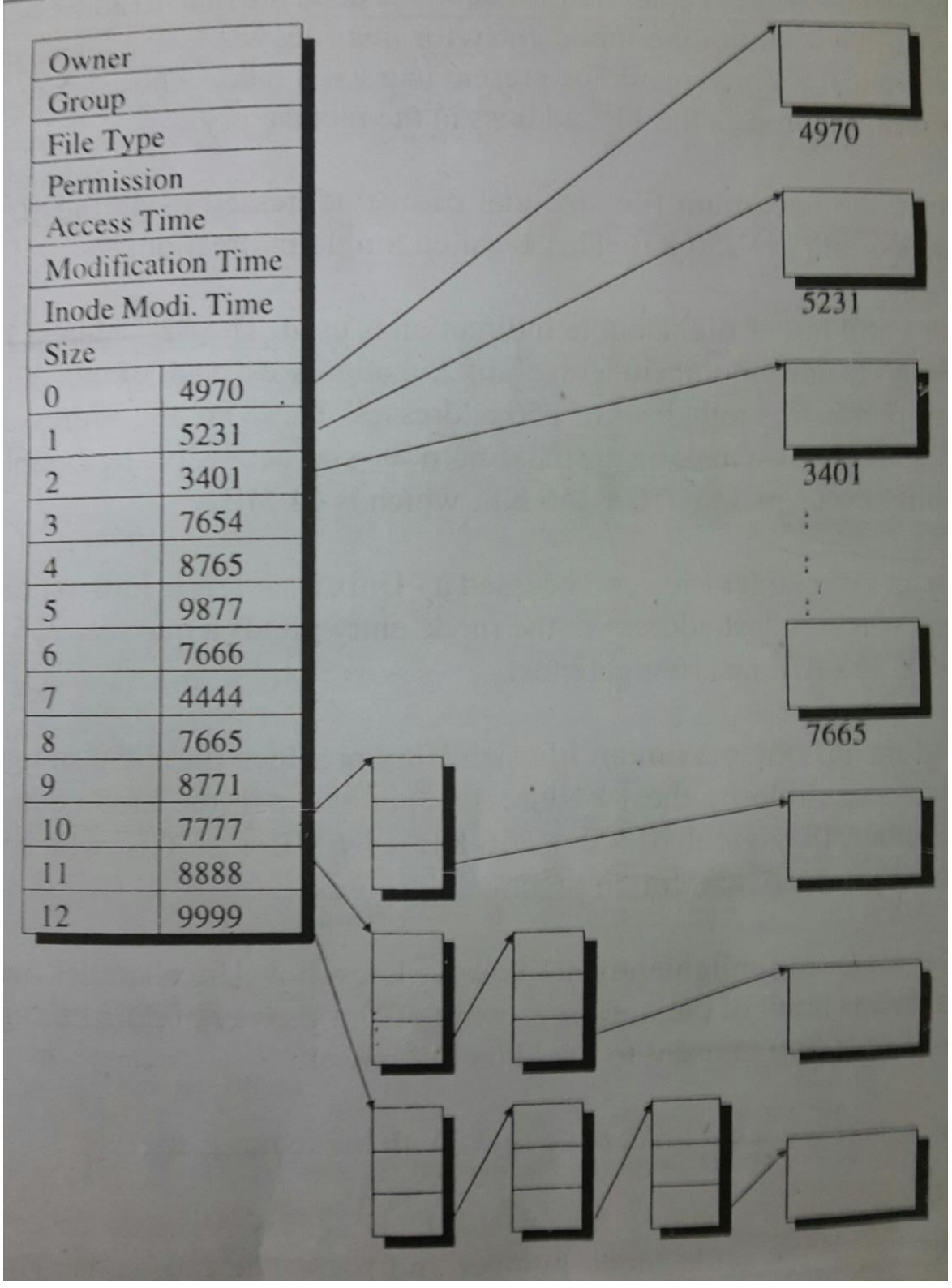
Suppose file 'reports' is present in 'mydir' directory and we attempt to cat the 'reports' file:

1) Check whether mydir have read permission
2) Find out whether this directory file has an entry “reports” in it
3) Pick up the inode number for “reports” from mydir
4) This inode number is an index into the in-core (memory) inode table
5) Using this inode number, information about reports can be accessed from the inode table
6) From this information, it is found that whether we have a read permission for the reports file
7) Then contents of the file are read from the disk addresses mentioned in the inode entry of reports and displayed on the screen

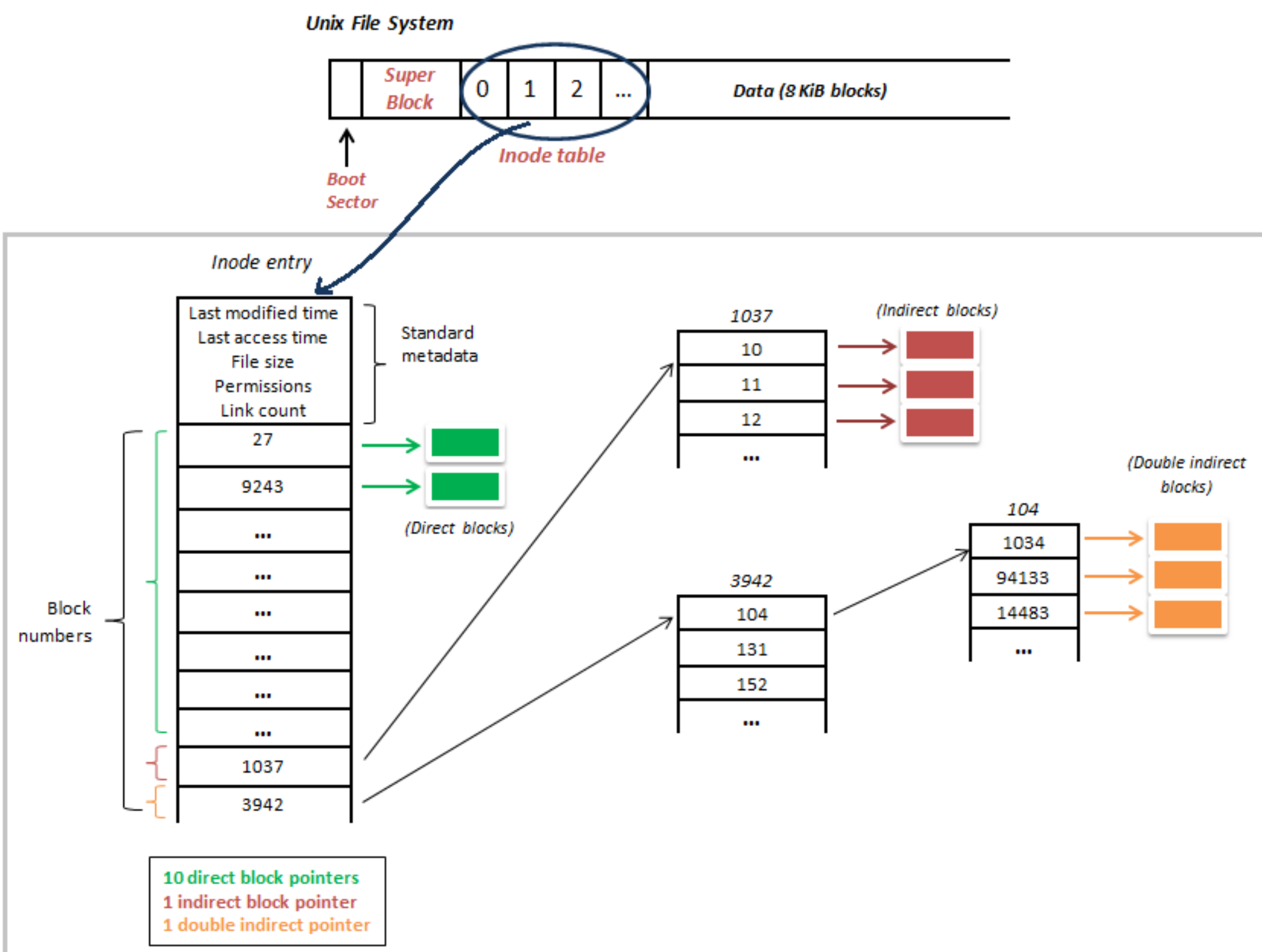

ext2_inode
Mode
Owner info
Size
Timestamps
Direct Blocks
Indirect blocks
Double Indirect
Triple Indirect
Data
Data
Data
Data
Data
Data
Data
Data

*A file of size 3KB may have its entries as*






*For Files larger than 10KB*


Unix File System
Super Block
0
1
2
...
Data (8 KiB blocks)
Boot Sector
Inode table
Inode entry
Last modified time
Last access time
File size
Permissions
Link count
Standard metadata
27
9243
...
...
...
...
...
...
1037
3942
Block numbers
(Direct blocks)
1037
10
11
12
...
(Indirect blocks)
3942
104
131
152
...
104
1034
94133
14483
...
(Double indirect blocks)
10 direct block pointers
1 indirect block pointer
1 double indirect pointer