KOÇ UNIVERSITY

COMP304 - PS
Simple File System Implementation

Najeeb Ahmad

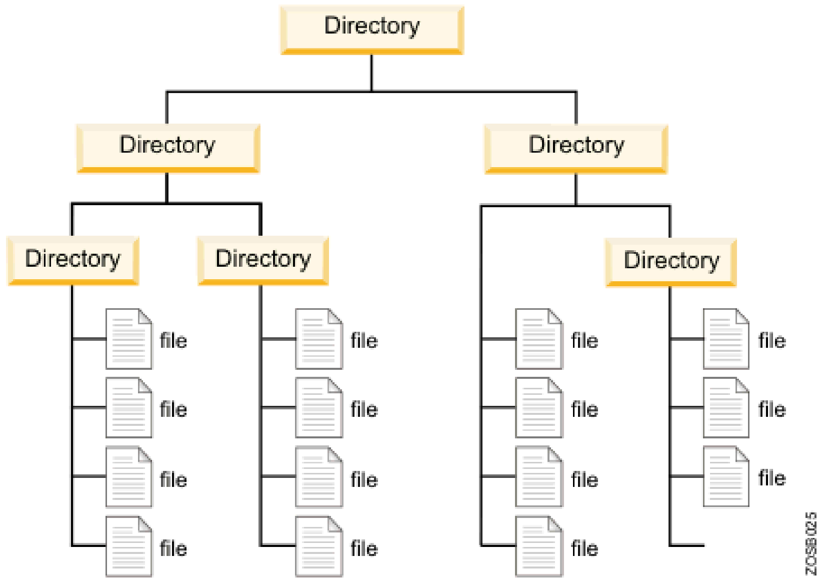
Koç University, Istanbul, Turkey

PS Outline

- Overview of file system
- Overview of the project

File System

- File system helps organize and retrieve files on storage media
  - Data organized in files and directories
- Examples of file systems:
  - Windows: FAT, NTFS
  - Apple: APFS

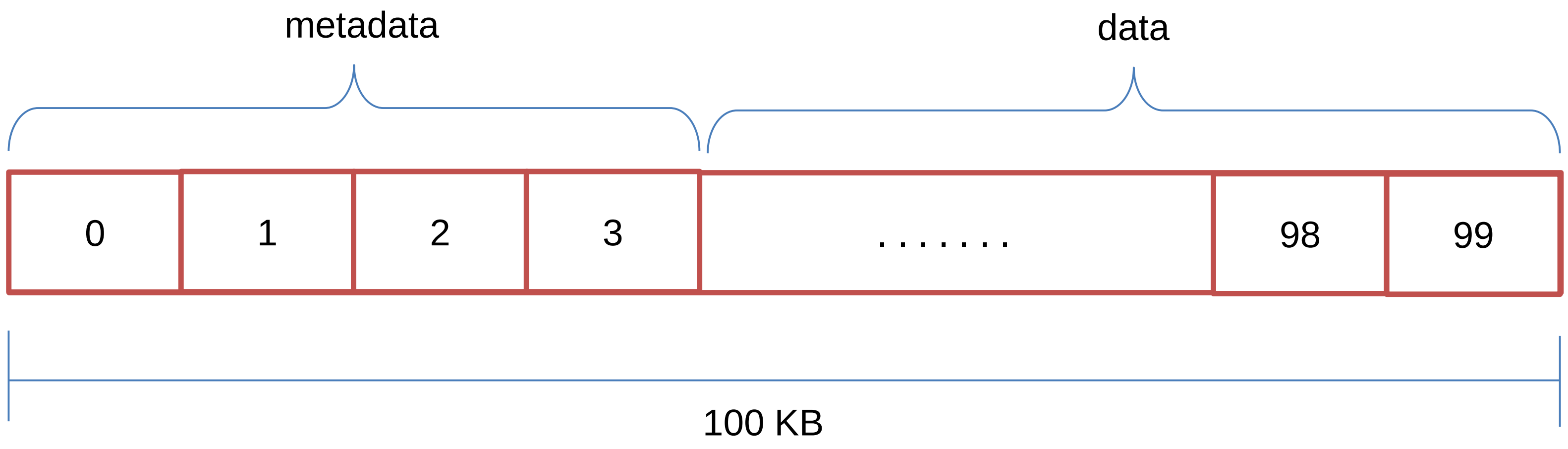


Project Overview

- To help you understand how file systems are implemented
- You will be working with FS304
  - Emulated on file fs304.disk
    - 100 KB in size (provided)
    - Mimics a disk
    - All file system related operations will be performed on this disk file

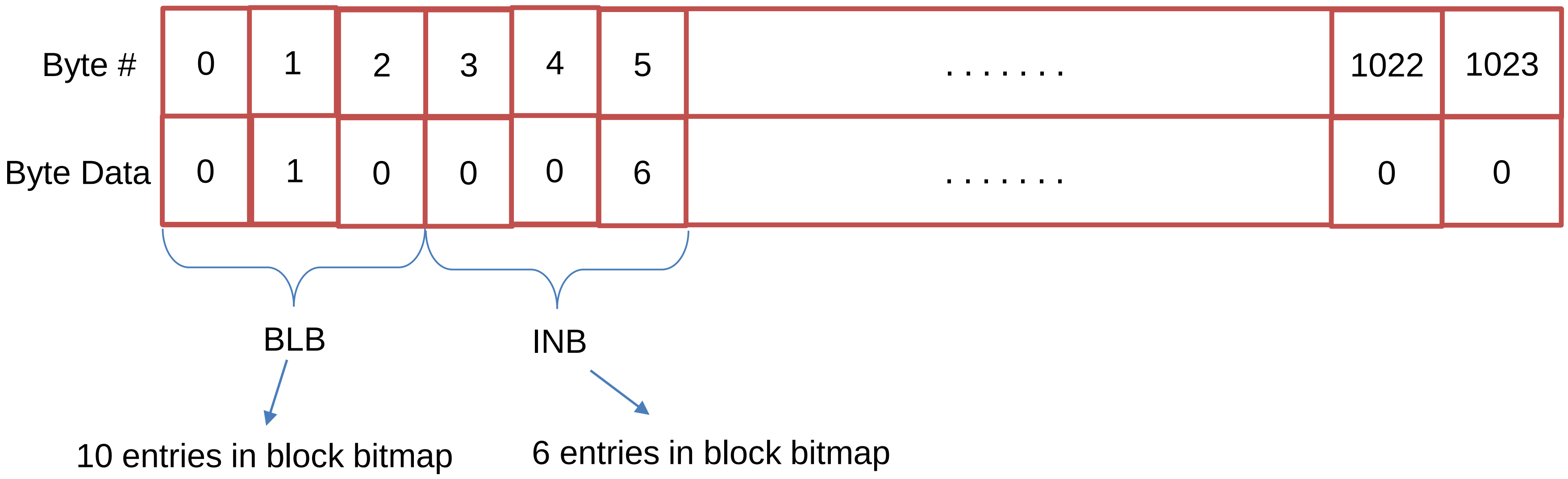
FS304 Organization

- Data on fs304.disk divided into blocks of length 1024 bytes each (100 blocks)
  - Size is fixed. Should not be modified.



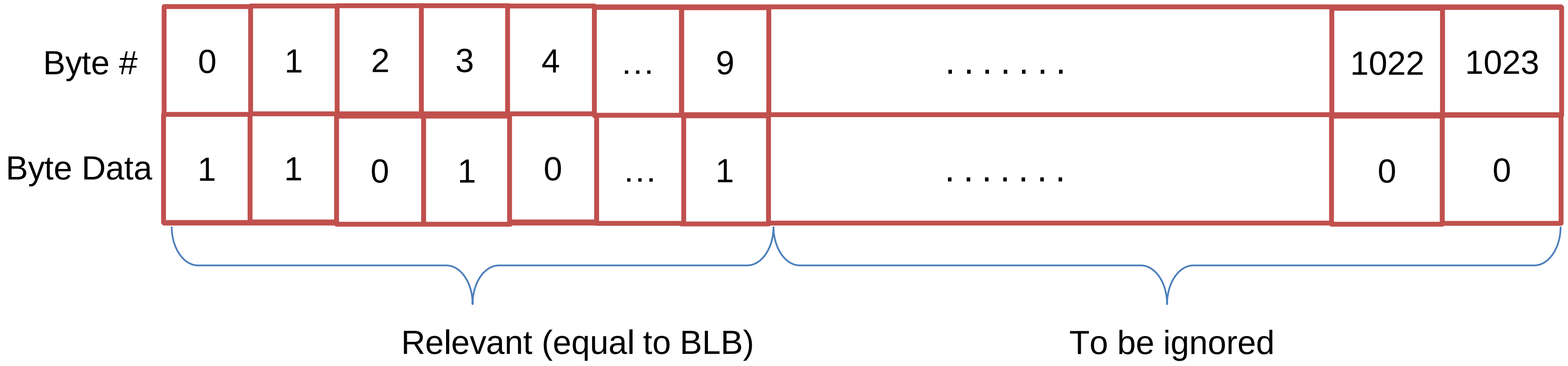
FS304 Metadata

- Block 0 (Superblock)
  - 6 bytes contain useful data
    - 3 for number of entries in block bitmap (coming up)
    - 3 for number of entries inode bitmap (coming up)



FS304 Metadata

- Block 1 (Blockbitmap)
  - Total 1024 entries
  - Only BLB number of entries are relevant
- 0 at *i*th index indicates block *i* is available. 1 indicates it is unavailable

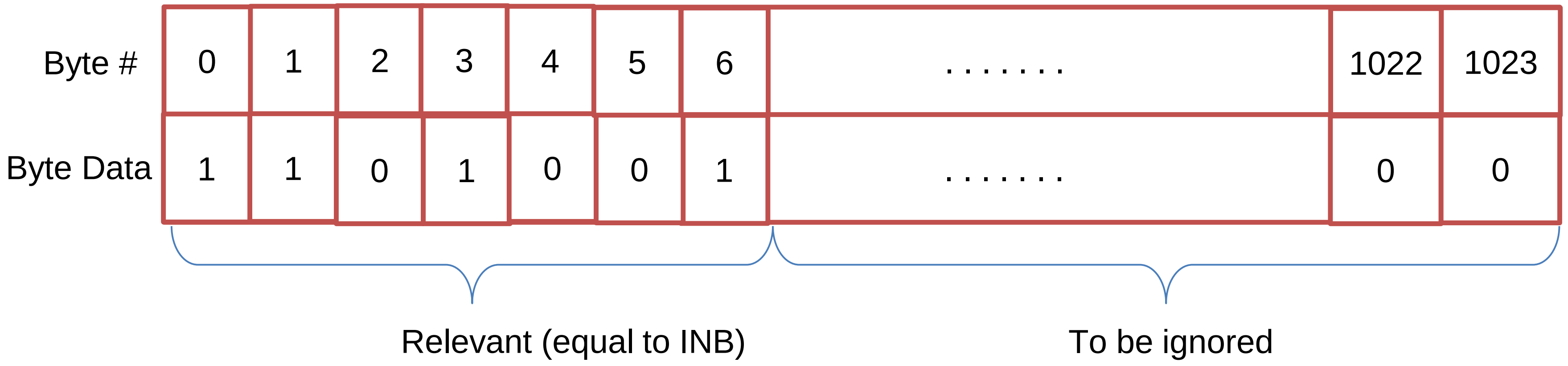


Byte #	0	1	2	3	4	…	9	. . . .	1022	1023
Byte Data	1	1	0	1	0	…	1	. . . .	0	0

Note: These relevant and ignore figures are with reference to example on previous slide

FS304 Metadata

- Block 2 (inode bitmap)
  - Total 1024 entries
  - Only INB number of entries are relevant
- 0 at *i*th index indicates inode table entry *i* is available. 1 indicates it is unavailable



Byte #	0	1	2	3	4	5	6	. . . .	1022	1023
Byte Data	1	1	0	1	0	0	1	. . . .	0	0

Relevant (equal to INB)
To be ignored

Note: These relevant and ignore figures are with reference to example on slide for block 0

FS304 Metadata

- Block 3 (inode table)

Byte #	0	1	2	3	4	5	6	7	. . . .	1022	1023
Byte Data	T	T	X	X	Y	Y	Z	Z	. . . .	0	0

$\underbrace{\text{Bytes 0–1}}_{\text{FI Or DI}}$ $\underbrace{\text{Bytes 2–7}}_{\text{IDs of three blocks storing file/directory contents}}$

FI
Or
DI

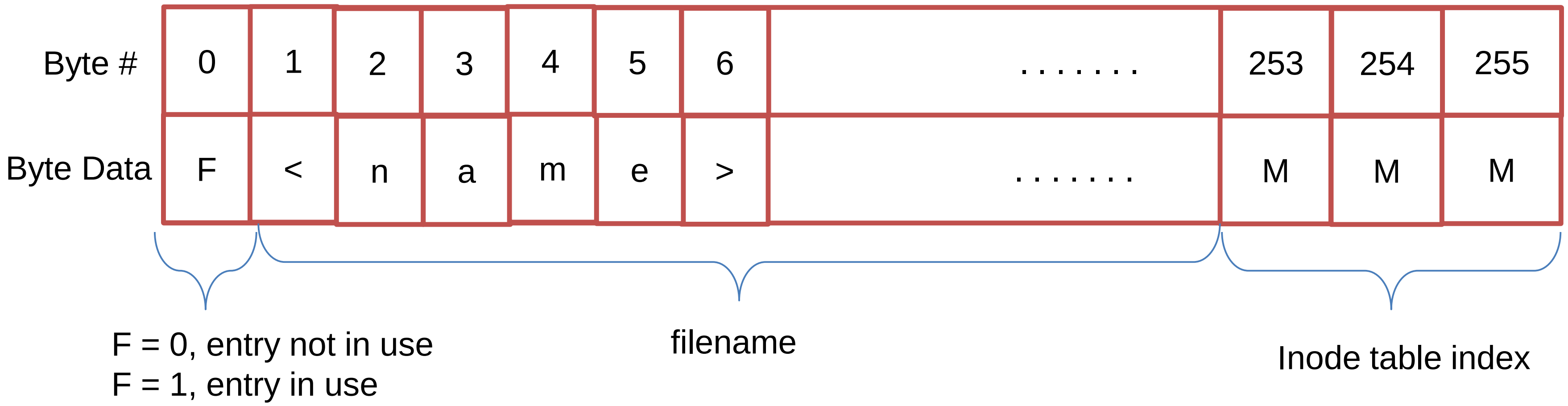
IDs of three blocks storing file/directory contents

XXYYZZ = 000000 means empty file
Each of XX, YY, ZZ vary between 04 - 99

- First entry always for root

Directory information

- If TT = DI, XX, YY, ZZ store directory info in the following format



- We can have 12 entries per directory

Existing functionality

- fs304.h and fs304.disk provided. Following functions are already implemented:
  - ls: List files and directories in current directory
  - cd <dir>: Change directory
  - md: Make directory
  - rd: Remove directory
  - stats: Prints free inode entries and blocks

Helper functions

- Following helper functions are provided:
  - `int readFS304(int BN, char buf[1024])`
  - `int writeFS304(int BN, char buf[1024])`
  - `mountFS304()`
    - Make sure to call this before performing any of file system related tasks

Required Implementations

- Implement file compare function
  - `compare <filename1> <filename2>`
- Files to be compared will be in the current directory
  - You can check how `ls` command finds files in current directory
  - Then check if files with given name exist
  - Finally read contents of each file using readFS304() function and compare

Required Implementations

- Implement file rename function
  - `rename <oldname> <newname>`
- File to be renamed will be in the current directory
  - Again, you can locate file in a similar manner like ls command
  - Make sure file with oldname exist
  - Rename file in F<name>MMM entry for the file

Required Implementations

- Implement file copy function
  - `copy <file1> <file2>`
- File will be copied into current directory
  - First a new file `file2` will be created
    - First check if file with same name exists. If so print error
    - Also check if space is available for the file. Else print error.
    - Check `md` command for help on file creation
  - Read `file1` using readFS304() function, write contents to `file2` using writeFS304()

Suggestions

- Use fs304.h to learn implementations of existing commands to help you with your implementations
- For the copy command, its better to implement create function separately (in case you just want to create new files)

THANK YOU