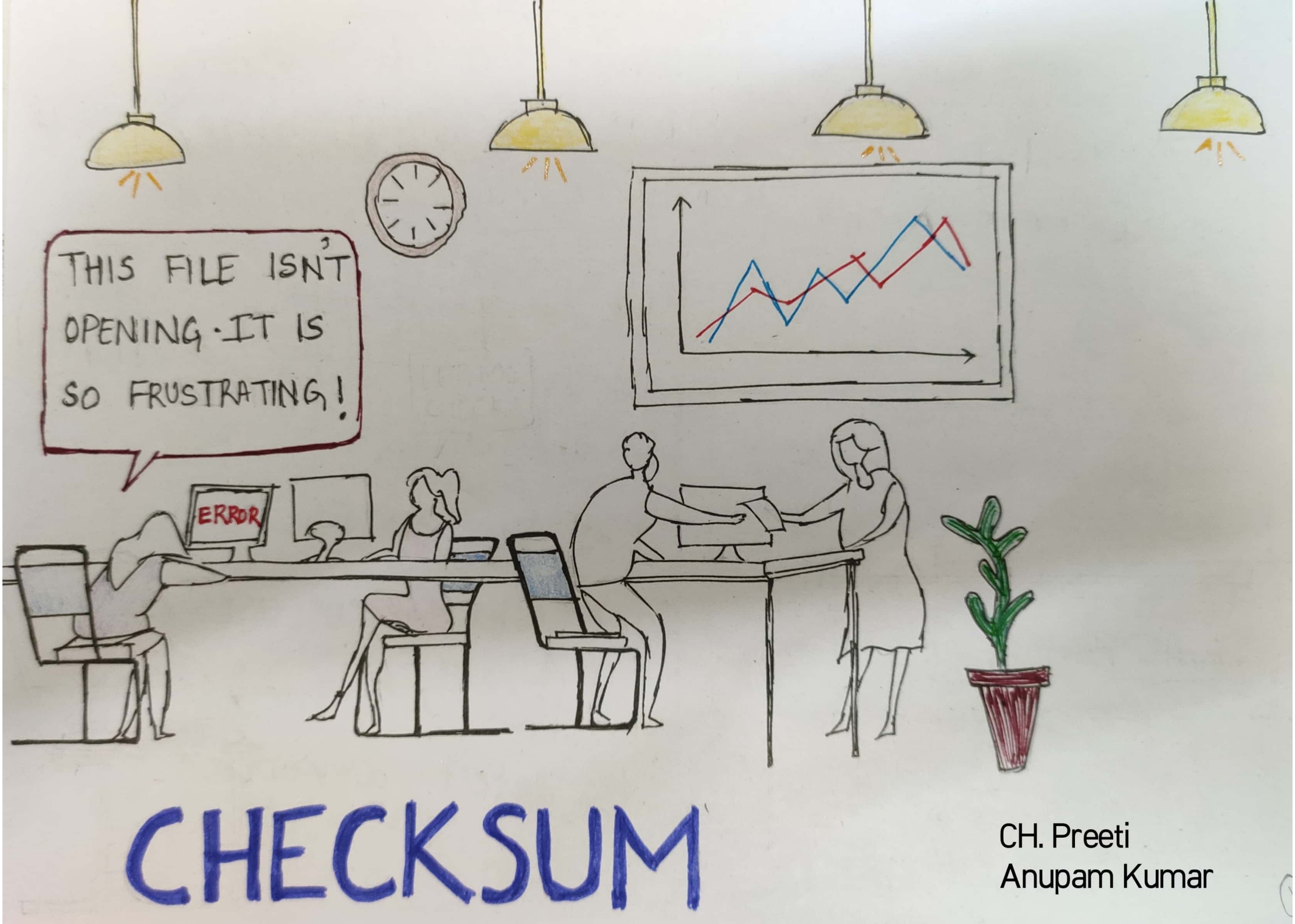
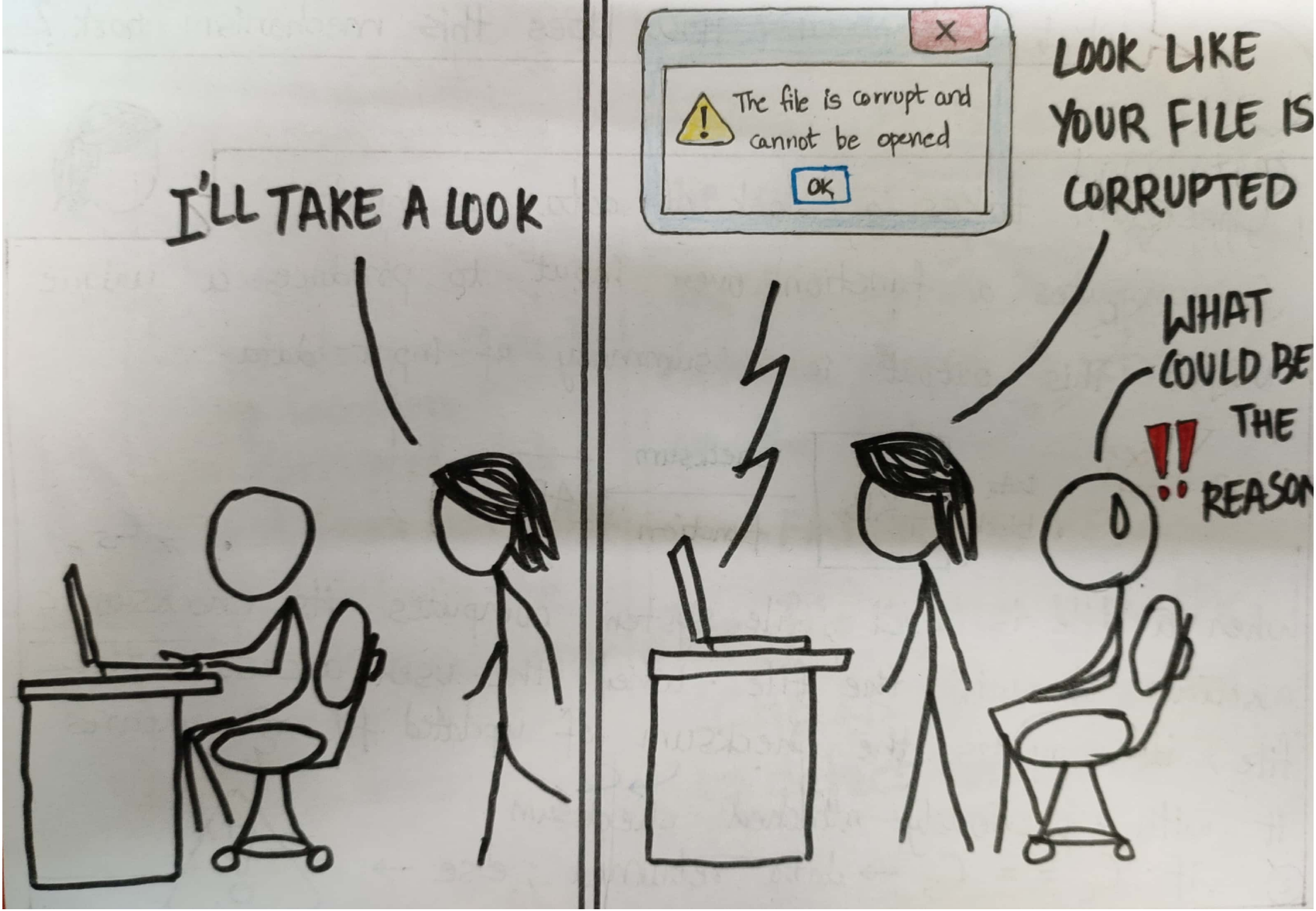
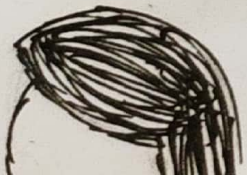




a file gets corrupted while reading, writing, storage or processing of data :

- correct data written to wrong location
- data corrupted while transfered from host to disk

File system detects the faulty data & throws error.



what is a file system & how will it detect the corruption ?

The file system controls how data is stored & retrieved on the disk

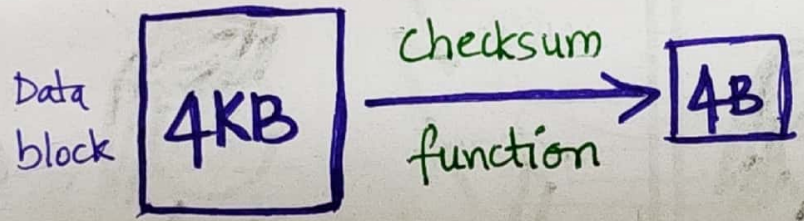
Ex: FAT, NTFS, HFS

The file system uses the CHECKSUM mechanism to detect corruption

What is checksum? How does this mechanism work?

Checksum takes a block of data as input & computes a function over input to produce a unique output. This output is a summary of input data.

Ex:

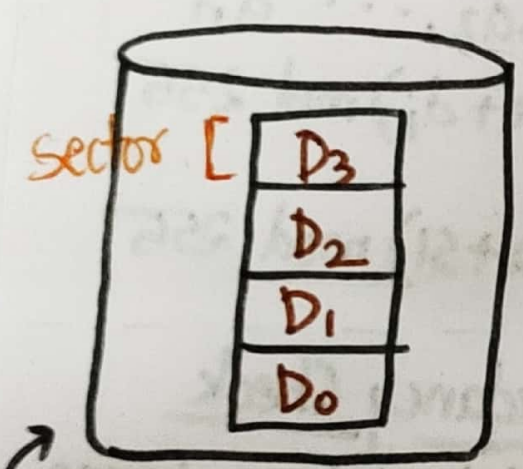


When a file is saved, file system computes its checksum (C_s) & associate it with the file. When the user accesses the file, it computes the checksum (C_c) of updated file & compares it with previously attached checksum

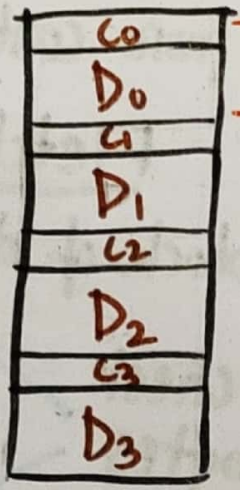
if $C_c == C_s$ → data returned; else →

How is Checksum stored on disk?

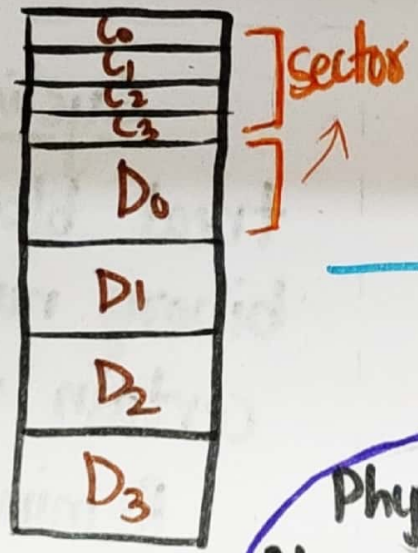
D_i = blocks of data



Disk without checksum



Size of $D_i + C_i$ = multiple of original sector size



combining all checksums to form a new sector of same size as D_i

Disks only can write in sector-sized chunks or multiples of it.

Physical identifier :
Along with checksum, we store sector no. & disk no (if multiple disks)

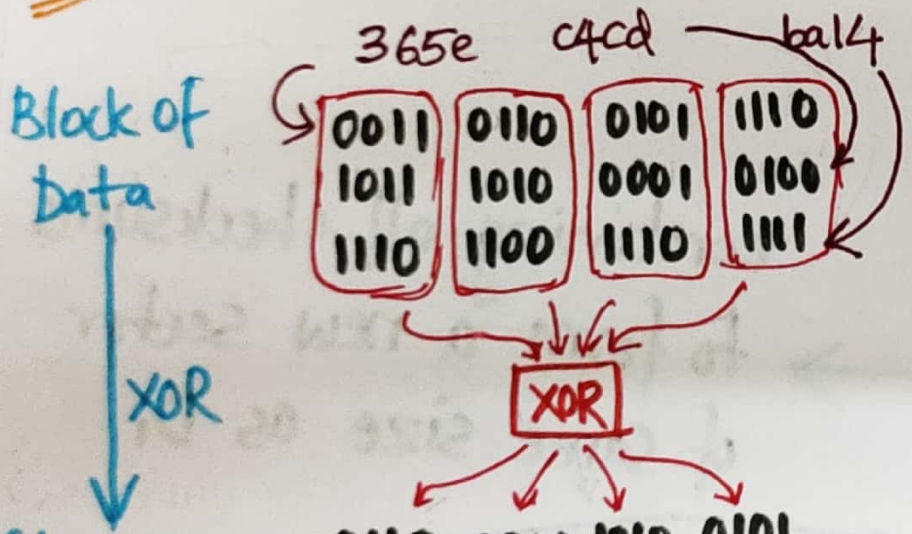
Types Of Checksum Functions

NO FREE LUNCH : more protection ⇔ more costly (in strength & speed)

XOR

XORing parts of data block

Ex: block of 4 bytes in hex



ADDITION

Adding chunks of data

✓ Fast

Fletcher Checksum Algo.

block of data: $d_1, d_2, \cdots d_n$

2 check bytes

- $S1 = (S1 + d_i) \bmod 255$
- $S2 = (S2 + S1) \bmod 255$

Cyclic Reducdancy Check

treat block of data(D) as large binary number. Divide it by certain value k.

Reminder (D, k) = CRC

✓ implementation of modulo operator is easy

BUT WILL THESE FUNCTIONS ALWAYS GIVE UNIQUE OUTPUT?

No! There are execptions when then output is not unique

Collision : Same checksum for corrupt data & original data

Ex:

✗ If 2 bits in same position within each checksummed unit change → COLLISION in XOR function

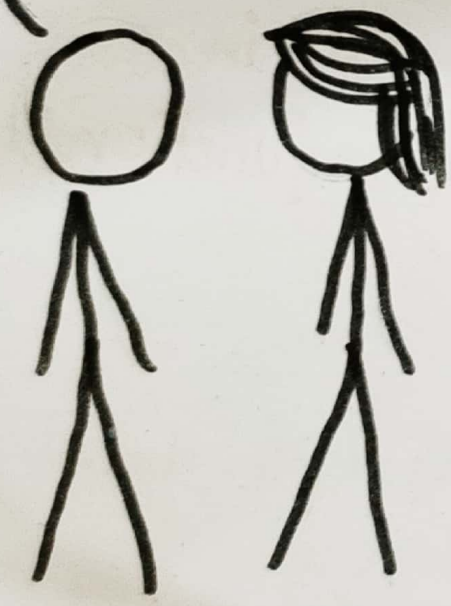
✗ If block of data is shifted → COLLISION in addition function

CHECKSUMS ARE CHECKED WHILE FILE SYSTEM IS ACCESSING A BLOCK OF DATA OR DURING DATA SCRUBBING

↓

Checking checksums of all files in memory

WHEN DO THESE CHECKSUMS GET CHECKED?



No! Sometimes the storage informs that a write has completed when it is still persisting. In such cases, data isn't updated. Such failure is called lost writes. These can't be solved by checksums

Does checksum always detect data corruption?

Overheads of Using Checksum

Space

extra space on disk to store checksums for blocks of data

Ex: 8 byte checksum for 4KB datablock cause 0.19% disk overhead

Time

extra time to compute checksums

- data is stored (C_s)
- data is accessed (C_a)

and compare the two

Hard drives are very sensitive &, we never know when they fail. The Checksums are just used to detect corruption. To recover data, we need to print a version of data (backup). In case of RAID system, automatic backup on deteting corruption