24th Dec 22

Linux
Filesystem, Partitions & Inode

Type characters

d -> directory
p -> pipe
- -> file

l -> link
c -> character
b -> block

same as pipe but can handle more than 2 process at once
p -> pipe
s -> socket

a character device -> which passes / shares data char. by char

block device transfer data in big size of blocks (ex -> HHD, SSD)

you can check which device file is getting used by -> ls -l /dev -> directory

The devices can have different partitions ex

nvme0n1p1
nvme0n2p2

pseudo devices -> which are not physically connected to system

/dev/null -> char devices
/dev/zero <- examples

/sys directory -> gives more detailed info about /dev directory

/dev allows programs to access devices whereas /sys is used to view information & manage devices

/sys directory is a virtual filesystem provided by Linux
↳ contains all details/info about devices

Filesystem Hierarchy

① / -> root directory
② /bin -> contains essential & ready to run binaries like ls
③ /boot -> contains bootloader files
4) /dev -> contains device files
5) /etc -> contains configuration files
6) /home -> home directory
7) /lib -> contains libraries
8) ~~/media~~ ->
8) /mnt -> temp mounted FS
9) /opt -> optional software packages
10) /proc -> process info
11) /root -> home directory for root
12) /sbin -> system binaries -> can only be runned by root

Journaling -> used to repair any inconsistencies that occur as the result of an improper shutdown of a computer

Suppose you were copying a file & system shutdowns so when booting up & the system will chick the journal to remove corrupt files

Desktop FS types

1) ext 4 -> latest & standard choice of a FS
   - support disk space -> 1exabyte
   - support file size -> 16 Tb

2) Btrfs -> Butter/Better FS
   - not stable as of not

3) XFS -> high performance journaling FS
   - generally good for servers that hosts media or something

4) NTFS & FAT
   - windows FS

5) HFS+
   - MAC FS

to check your FS -> df -T
(note -> partitions can't overlap)

We can create multiple partitions in any disk & those each partition will act as an individual block device
   - then we can have different FS

Partition Table -> To check how a disk is partitioned

There are two main partition schemes
① MBR -> Master Boot Record
② GPT -> GUID Partition Table

→ If we have some space which is not a part of any partition then that is generally known as free space

MBR - traditional partition table, supports disk upto 2TB
→ has limitations
→ of 4 parts only
→ known as primary partitions
out of these 4 we can create 1 extended partition
→ & in that can create multiple logical partitions — as same works

GPT → new standard
→ each partition has Globally Unique ID (GUID)
→ usually used with UEFI based booting

Filesystem Structure

FS is just an organized collection of files & directories, it's like ~~an~~ a database to manage files &

FS 4 Major Components

- Boot Block
- Super Block
- inode table
- Data Blocks

Inode
↳ Index Node

Inode Tables are just like database to manage files
→ in this table each file or directory has inode & it generally describes all the info about file

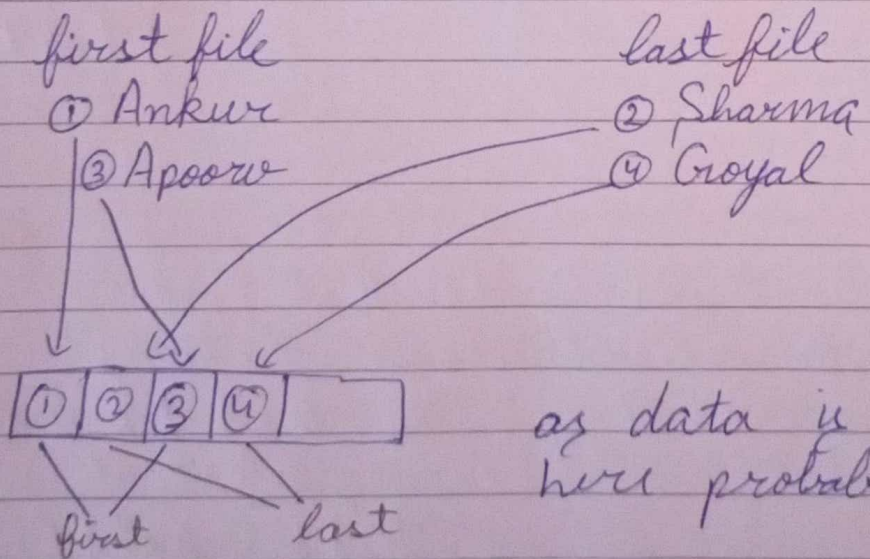
↳ Contains everything except for file's id's name, also contains pointers to datablocks of file

When ~~we~~ FS is created → some space for inodes is allocated as well

Command to check how ~~much~~ much inodes are available
↳ df -i

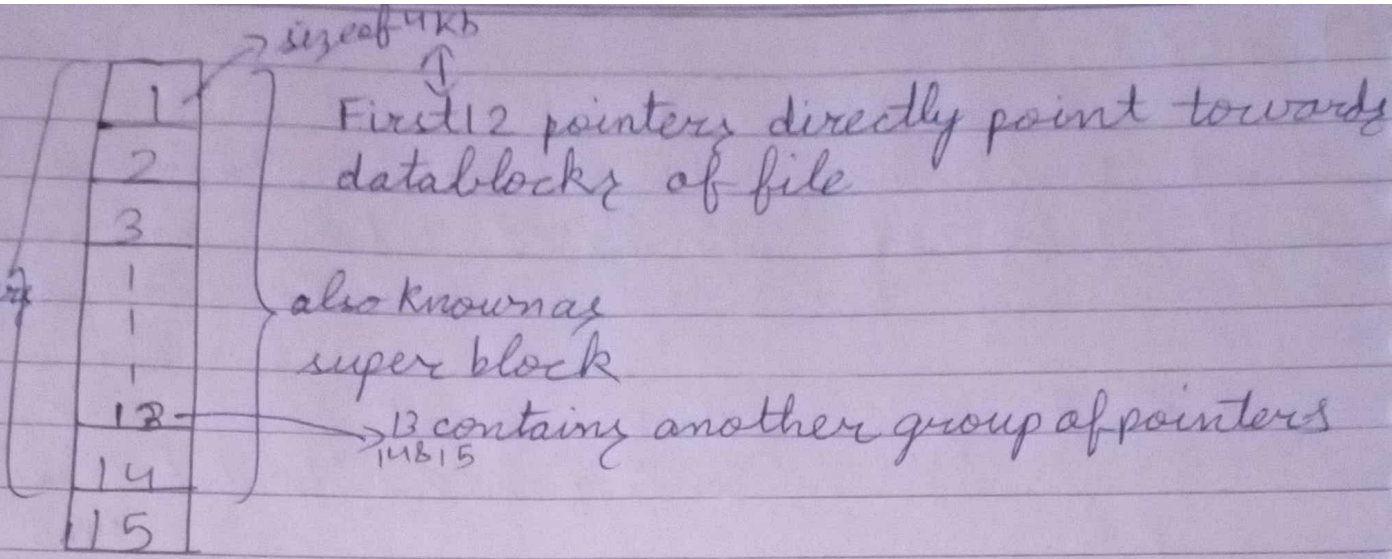
checking inode no → ls -li

How do Inodes work & locate file

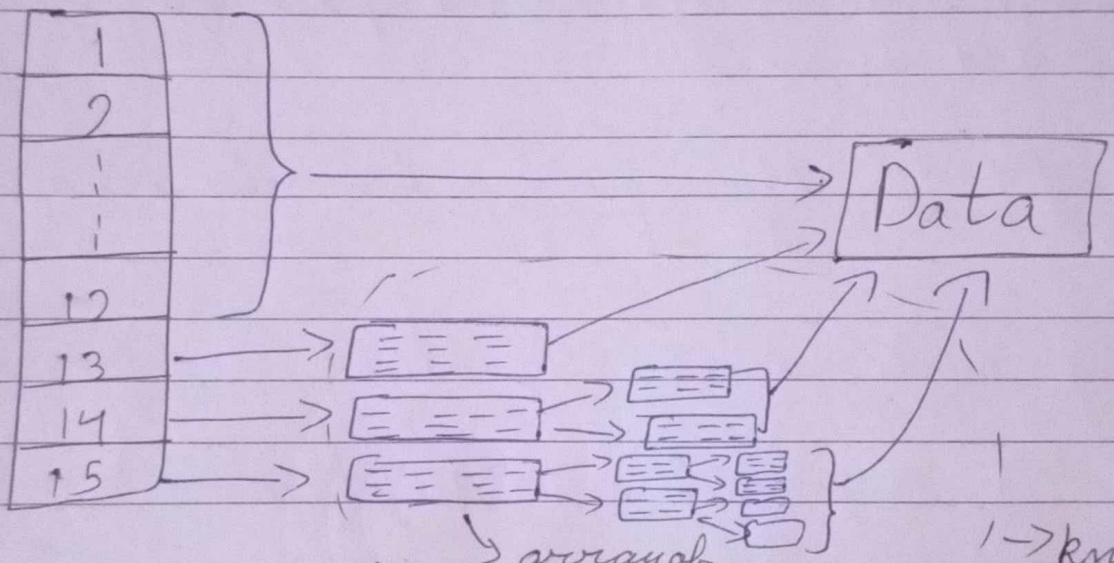


as data is not stored sequentially here probably occurs sometimes

so to find all data we use Inode
inode points to actual datablocks of file
in a FS each & every inode contains 15 pointers



↓ so it looks like



because of this max. size of a file can be 16 TB

Symlinks / Symbolic links / Soft Links
→ equal to shortcuts in windows
suppose you have created a file test1 & wants it's shortcut
→ command ln -s test1 soft1
(-s for soft link; test1 → file name; soft1 → shortcut name)

Hardlinks → command → ln test2 hard2