

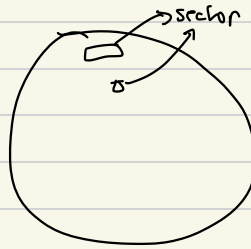
# Files and Filesystems

## File Abstractions

- storage devices needed
- hardware w/ registers
- linear area of blocks
- specifying id

## 5 orders slower

CPU  $1000\times$   $\leftarrow$  DRAM  $1000\times$  Disk



1 ... 0xFFFF...

## only has read or write

## need to remember block #

## coredump

- ↳ memory was initially a magnetic coil
- ↳ same as memory dump

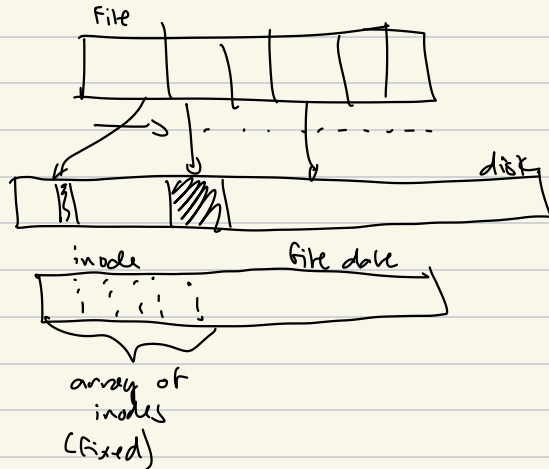
## file: named collection of data

## persistent across reboots

## metadata v. data

## inode

- ↳ metadata describing the file



stat: dumps inode info

## fsync

- ↳ force modification to disk

## everything is a file

## file: named collection of data

## directories: hierarchal name space

## security

## tolerance of crashes and failures

- ↳ each inode talks about a specific file

- ↳ freed when a file is removed

- pipe  $\Rightarrow$  is a file
  - $\hookrightarrow$  comes from the buffer

## File Abstraction

- inode: piece of metadata that represents the data
- only certain time will data come from disk
- `ioctl()`  $\Rightarrow$  another interface to device file
- directories
- mounting points to the root
  - $\hookrightarrow$  builds connections from filestructures to inodes
- Loop device
- Links
  - $\hookrightarrow$  hard link
  - $\hookrightarrow$  soft link
- ref cnt  $\Rightarrow$  keep track of
- data deduplication
- soft link  $\Rightarrow$  textual string of a's path
- Permissions
  - $\hookrightarrow$  fork  $\rightarrow$  exec  $\rightarrow$  call kernel
  - $\hookrightarrow$  <sup>parzing</sup> matching inode  $\rightarrow$  parses to create elf file  $\rightarrow$  close  $\leftarrow$  free all passed  $\rightarrow$  exit  $\rightarrow$  cain
  - $\hookrightarrow$  shell is sleeping

directory is a file

inode #	name

• permission

- kernel has to enforce permission
- groups are like tags
- based on userid, kernel chooses who can do what
- File Access Permission
  - ↳ 9 permission bits
- For directories
  - ↳ R: list files
  - ↳ W: create new files/subdirs in dir
  - ↳ X: traversal
    - ↳ does NOT imply R

• program can run on behalf of another user

- ↳ setuid

- when creating a new file → look for empty block in MFT
- root needs to be hardcoded
  - ↳ always one file even w/ empty machine
- each file is linked list
- protocol held by code in kernel

Ext 2

- inode table
- ext2 inode

- metadata

- set of 12 datapoints

- indirect block pointer

- indirect block

• max file size = 4GB

$$10^3 \cdot 4 \cdot 10^3 = 4 \cdot 10^6$$

$$\text{offset} = \frac{\text{offset}}{\text{Bsize}} = \text{location of block}$$

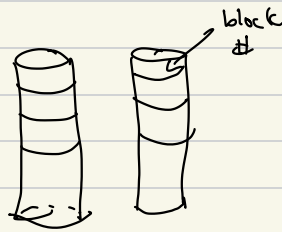
- ext2 locality
  - better for them to be closer
  - allocate blocks w/in group

- B+ tree as index

- Data Persistence
  - expects durability

## Redundant Array of Independent

- RAID 0: striping
  - motivation: speed
  - no redundancy
  - reliability worse than a single disk



- RAID 1: Mirroring
  - motivation: reliability
  - usable capacity reduced
  - write: all disks
  - read: any disk
  - if one drive goes down  $\rightarrow$  make a copy

## RAID 5

- can reconstruct any missing block

## RAID 6: tolerate 2 failed disks

- at least 5 hard drives

## File System Reliability

• ex.

- append new block to existing file
  - ↳ write in memory bitmap B