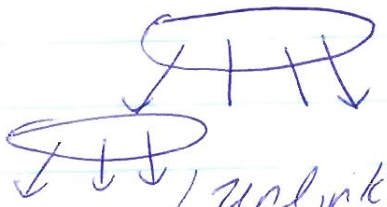
Data Structures.

User-level —



pathname
directory separators
one ..

unlink removes a file

API to search directory
(note not normal as system call —

Recursive!
Each file one —
~~hard~~ ..

n-ary tree

hard links
allow sharing of files..
Multiple names.

link allows a new name.
Restrictions → NOT to another file system.
~~they can't~~
NOT to a directory
Cycles..

Symbolic / soft links →
Seem to allow loops +
AND non-existent files..

A

Reference Counts — Link counts

# of names for a file.
Really delete only when link count is 0
AND file is not open
(can lead to mischief)

→ Numbers →
files are known by two numbers
(Device, inode #)
↑ index node

#2 is the Root!
#0 indicates - no file "DELETED file
#1 - once the bad block file

→ Mount -

"mounts" a file system on a directory (obviously Root is different)

A system call with lots of options

See /etc/fstab

(B) Ole examples of mount commands
/usr/local/sbin/netdump

"live system"

It's root is in memory, loaded from USB.

Other files are Mounted-

All operating systems have some convention.

/usr

\Program Files

"hidden" files ... on Linux

You need to know how to search files →

Unix find command..

Be to know how to program a search also

ARRAY -- fixed size

Disk may contains an array of inodes / File Control Blocks.

→ SSFS

Sequentially

or spread about

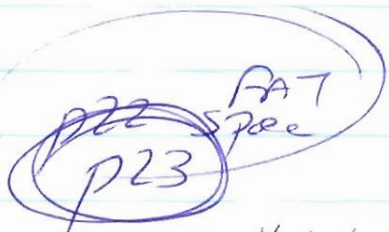
↑ info about the blocks.

OR organized logically - and dynamically

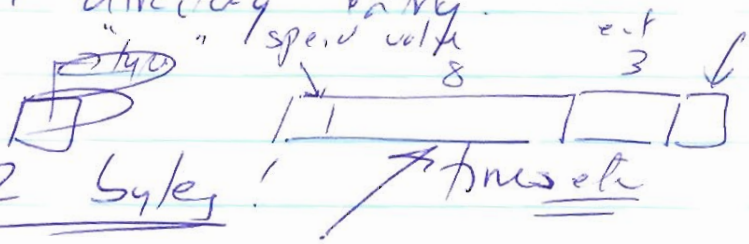
The "inode number" is the address of a file

ls -l <NTFS ->

Old-style DIRECTORS



FAT directory entry.

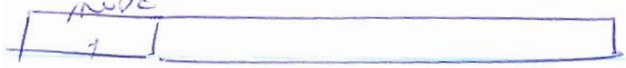


(E)

free block map (FAT)

System V FS

2 bytes — 14 file-name characters!



NOTE → in both FAT or S5FS.

entries are not ordered.

Sequential O(n) search!

Another simple array →
the BT ARRAY for allocated ~~blocks~~ blocks.

Generally kept in memory in a more efficient form to speed up allocation.

"linked list"

IN FAT directory entry, there's an address of first cluster

Address of cluster AND

FAT data structure → singly linked list of cluster

STARTS AT CLUSTER 2 — see page 13 of spec

Directory entry contained starting cluster

and starting FAT entry, FAT entryes give address of "next" cluster and next ~~file~~ FAT entry.

OLD-FASHIONED linked-list in FORTRAN

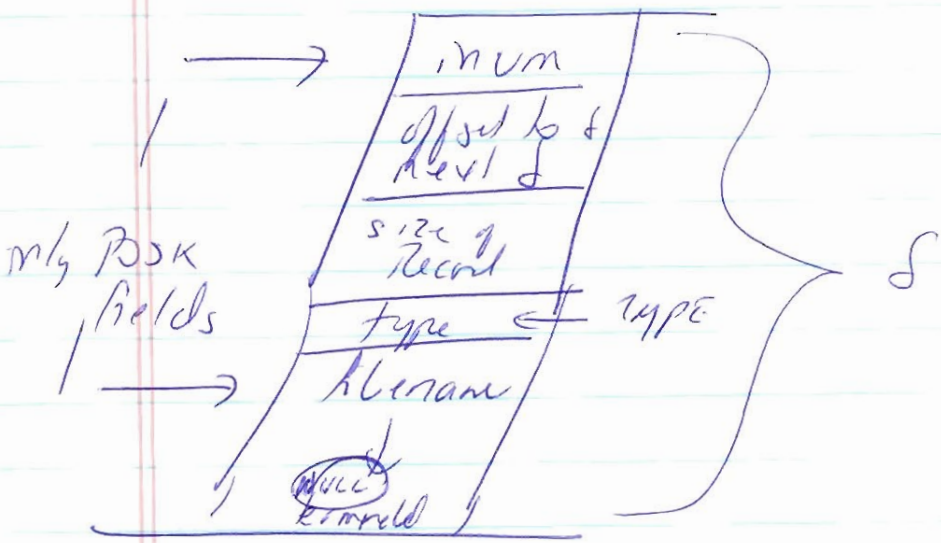
Access to Nth block

O(N)

6

Linked list FFS

man 3 readdir



Allow long file names --
FAT how is the partial part.

Diff - FAT + FFS.

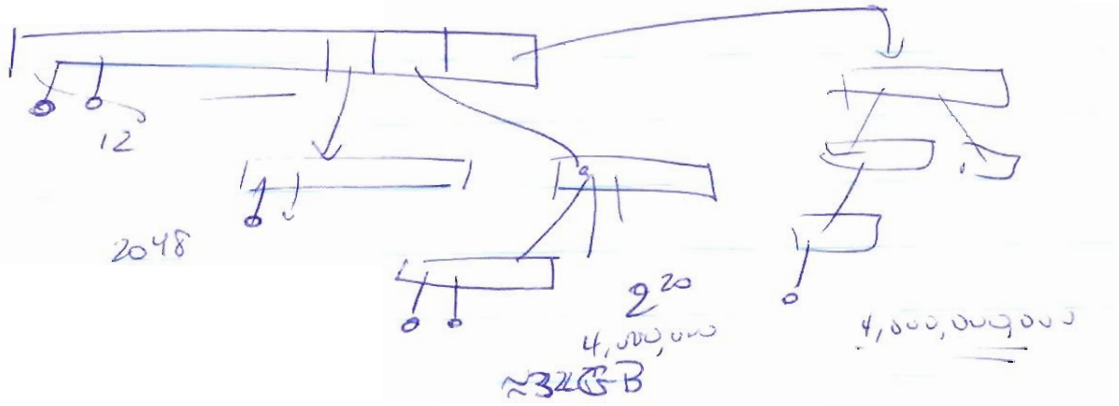
inode contains most file information.
owner, permissions, size, dates,
etc

An on-disk tree structure -

/usr/include/linux/ext2_fs.h

inode contain
up to 12 direct blocks
1 pointer to indirect
doubly indirect
tresly indirect

<4 "byte ext>
2,000,000,000



Access to Nth block in for mula
O (log N)

I

Problems.

block mapping
vs extent mapping

block-mapping →
in theory scatters data
in practice block map is sequential

extent map.
Represent file as a collection of
1 long data block.

=

Old FFS intentionally put parts of
long files into different
cylinder "groups".

w/o a declaration of file size —
what can be done?
Slow growing files are
particularly bad —

AdvFS, btrfs,
try to do extent mapping

5

Defragmenters try to reduce extents.
Rewriting a large file will
make it more efficient

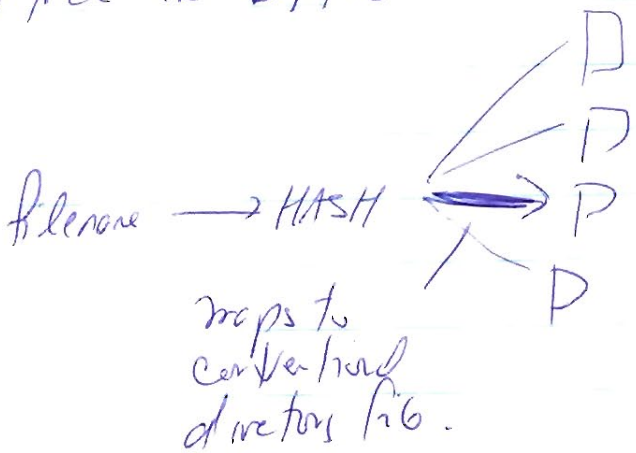
Large Directories are a problem -

Very slow search time -

Large directories used by
Web servers, browser, database
management system, git,

Solutions
B- trees as in database

H-tree in ~~btfrs~~



Logging →

Sometimes it helps to write more →

ext4, NTFS, Advfs, BTRFS

Allows file system to delay + schedule writes w/o a large danger of "lost" data.

Snapshots —

Preserve view of file system for user.

With C-O-W feature
M-O-W
Snow

LVM → volume manager at UNCA - CSCI