Systems 3

File Systems

Marcel Waldvogel

(Handout)

Department of Computer and Information Science
University of Konstanz

Winter 2019/2020

Chapter Goals

- What are the goals of file systems?
  - How did they evolve?
  - How do they work?
  - How are they used?
  - What data structures and algorithms are involved?
- How can reliability and data security be achieved?
  - What are the properties of backup and RAID?
  - How do they work?
  - How do they compare?
  - When is which needed?
  - What else is necessary?

Three views of a file system

- **Hardware view**
  Sectors, tracks, blocks, access time, disk scheduling ✓
- **Data view**
  Data structures, layout, properties
- **Application view**
  System calls, protection, programs

What is a file?

- From a user's perspective:
  - Persistent storage across reboots
  - A byte array
- From a OS perspective:
  - Secondary storage device
  - Map bytes as collection of blocks to storage device

How to determine file type?

- **File extension** (.exe, .jpeg, .txt)
  untrustworthy (extension can be changed by user)
- **File format** (e.g. magic numbers)
  untrustworthy (PHP image attack[1])

Question: Is the filename stored inside the file? Why (not)?

[1] "Programs do not care about file names. They do not even care whether the file has a name, or finally is it a file or something else!"
— "PHP: Running *.jpg as *.php or How to Prevent Execution of User Uploaded Files", Igor Data, Medium.com, 2017-01-24 (accessed 2019-12-05).

Example file formats

File type	Extension	Starts with
Unix script	—	'#!'
Unix ELF executable	—	0x7f 'ELF'
MS-DOS/Windows executable	.exe	'MZ'
SQLite 3.x database	.sqlite	'SQLite format 3.'
Graphics Interchange Format	.gif	'GIF87a' or 'GIF89a'
ZIP Archive	.zip, .odt, .docx, ...	'PK'
Tagged Image File Format	.tiff	'II.' 0x00 or 'MM' 0x00 '.'

See e.g. https://en.wikipedia.org/wiki/List_of_file_signatures

What is a file system?

Way to organize data in files and folders with meta data.
There are file systems for every use case (disk, flash, tape, database, network, ...), but the most common ones are:

- Disk (data formats):
  - (V)FAT (MS-DOS, Windows)
  - ext[234] (Linux)
  - NTFS (Windows)
  - HFS+/APFS (MacOS X, ...)
- Network (network protocols):
  - SMB
  - NFS

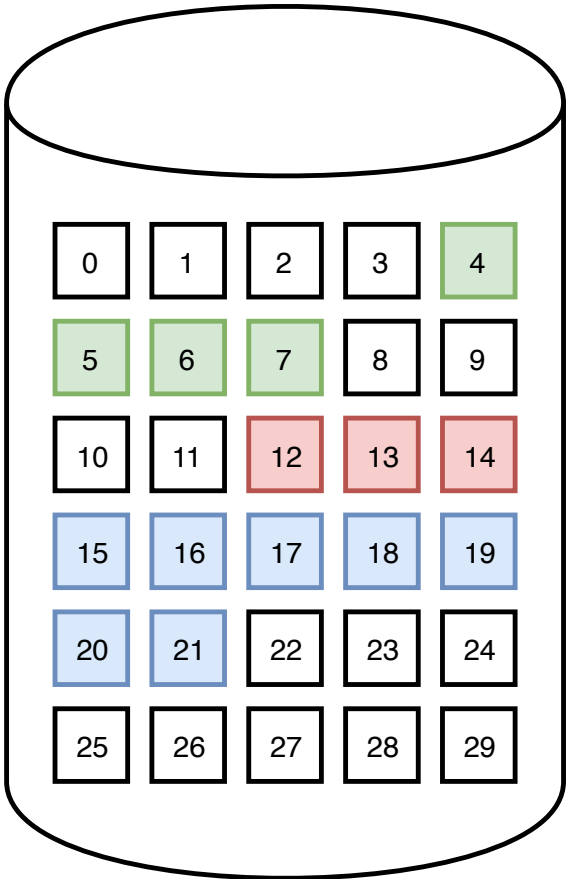
General file system layout

1. Boot control block (per volume)
2. Volume control block / superblock (per volume)
3. Directory structure
4. File control block (per file)

Allocation Methods

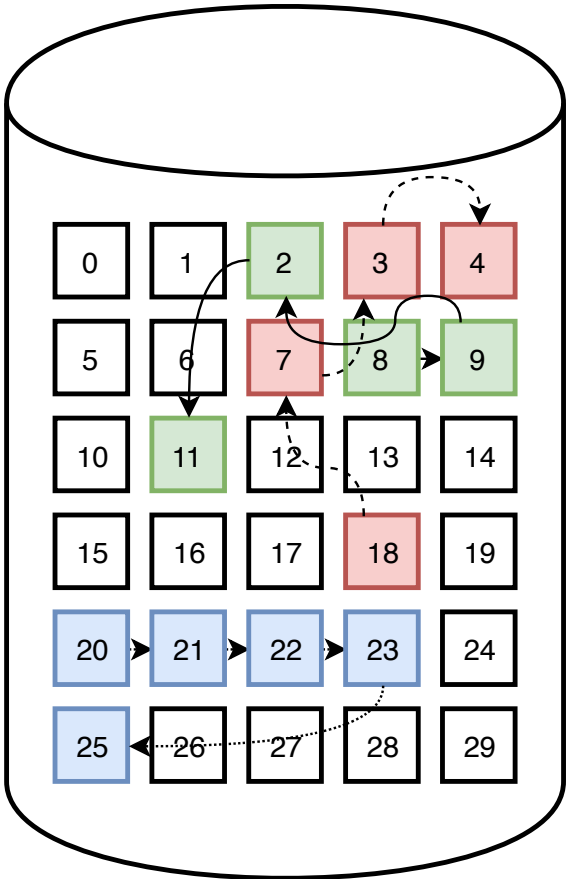
1. Contiguous Allocation
2. Linked allocation
3. Indexed allocation (linked, array-based, tree-based)

Contiguous Allocation



File name	Start	Length
slide.pdf	4	4
notes	12	3
logo.svg	15	7

Linked Allocation

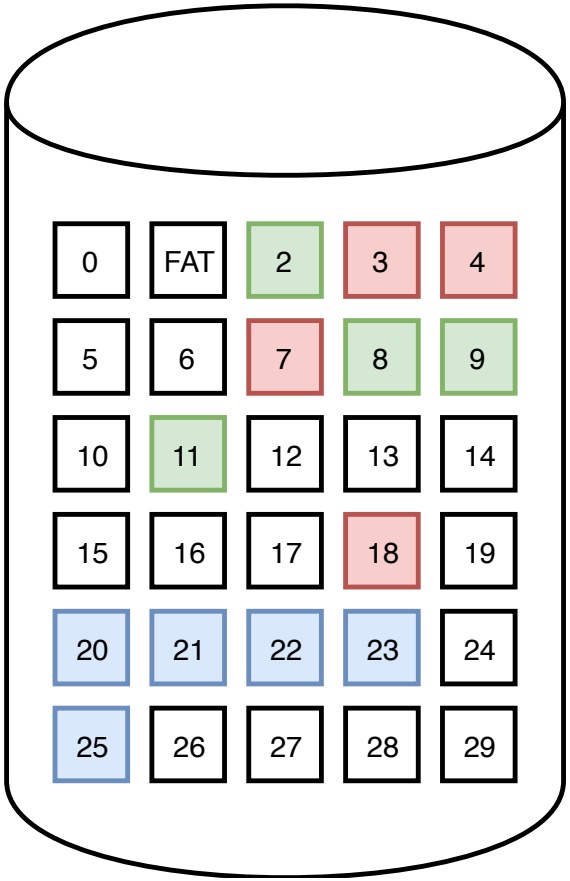


File name	Start
slide.pdf	8
notes	18
logo.svg	20

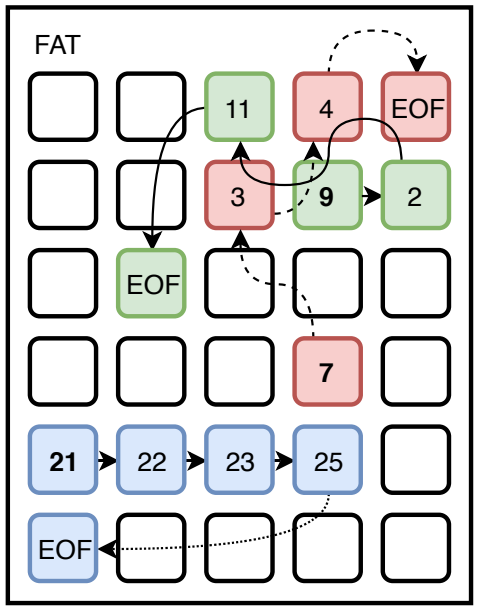
Usage: Many early systems: CP/M, home computers, ...

Linked list, but with seek latency.

FAT File (indexed linked)



File name	Start
slide.pdf	8
notes	18
logo.svg	20



Where to store file metadata?

Attributes in the directory entry.

Folder	Attributes
Documents	[attributes]
Pictures	[attributes]
Downloads	[attributes]
Music	[attributes]

e.g. MS-DOS, (V)FAT

Attributes elsewhere.

Folder	(inode) Pointer
Documents	6761391
Pictures	6767184
Downloads	6595408
Music	7595408

e.g. classical Unix

Question: What metadata is needed?

See also
https://en.wikipedia.org/wiki/Design_of_the_FAT_file_system
and https://en.wikipedia.org/wiki/Unix_filesystem.

Inodes

Contents:

- File mode (file type and RWX bits)
- User ID
- Group ID
- File size
- Timestamps (contents **m**odified, **a**ccessed; inode **c**hanged)
- Link count
- Block pointers
- ...

Question: Does the inode contain the filename? Why (not)?

See also https://en.wikipedia.org/wiki/Inode.

RWX bits

```
$ ls -l
total 16
drwx------ 5 klaus lecture 4096 Jun 18  2017 _ARCHIVE
drwxr-xr-x 7 klaus lecture 4096 Nov 29 17:54 exercise
-rw-r--r-- 1 klaus lecture 2251 Okt 31 12:00 README.md
drwxr-xr-x 2 klaus lecture 4096 Nov 29 17:55 slides
drwxr-xr-x 2 klaus lecture 4096 Dez  2 13:32 webserver
```

Access-Control Lists (ACL) can be used for more fine-grained permissions.
See e.g. https://linux.die.net/man/5/acl.

File vs. Memory

What is the difference between a file and memory access?

- Memory can be directly accessed, because every byte has an address inside a process
- Files are relative
  1. Not exclusive to a process
  2. Often accessed sequentially
  3. File can be moved from one directory to another
  4. To read/write a file a pointer (cursor) is used
- Files could be used by multiple processes (Problem?)
- A process is not allowed to read/write a file (Problem?)

File Operations

Metadata	Data	Misc
Create	Open	Seek
Delete	Close	Lock
Rename	Read	
Get attributes	Write	
Set attributes	Append	

Opening a file

The following steps are executed while you open a file:

- Resolve name
  1. Get starting directory (absolute/relative)
  2. Extract first path component
  3. Check for directory and enter
  4. Extract next path component, if any; continue at 3
  5. Last component may be file
- Read file metadata into open file table
- Return pointer (or index) to entry in open file table

Hard vs. Soft Link

```
$ ls -li
total 20
6767173 drwxr-xr-x 7 klaus klaus 4096 Nov 29 17:54 exercise
6707002 -rw-r--r-- 2 klaus klaus 2251 Okt 31 12:00 README
6707002 -rw-r--r-- 2 klaus klaus 2251 Okt 31 12:00 README.md
6761488 lrwxrwxrwx 1 klaus klaus   10 Dez  4 13:15 server -> webserver/
6767600 drwxr-xr-x 2 klaus klaus 4096 Nov 29 17:55 slides
6767676 drwxr-xr-x 2 klaus klaus 4096 Dez  4 13:14 webserver
```

Questions:

What are use cases for links?

What is the most common use case for hard links?

File System Reliability

File systems are trying to store information reliable, but they can't protect against:

1. Acts of God
2. Hardware or software errors
3. Human errors
4. Malicious software

Solution: Backups

Backup Issues

1. Backup all or part of the system?
2. Backup file if not changed?
3. Compression of backup or not?
4. Encrypt backup or not?
5. Backup while file system is active?
6. Physical security of backup media?
7. Choice of physical media?
8. File-based or image-based?
9. Incremental or full?
10. Where to store the backup?
11. Server or client? Business or personal?

Advantages? Disadvantages?

"Backup is never the problem. Restore is the problem."

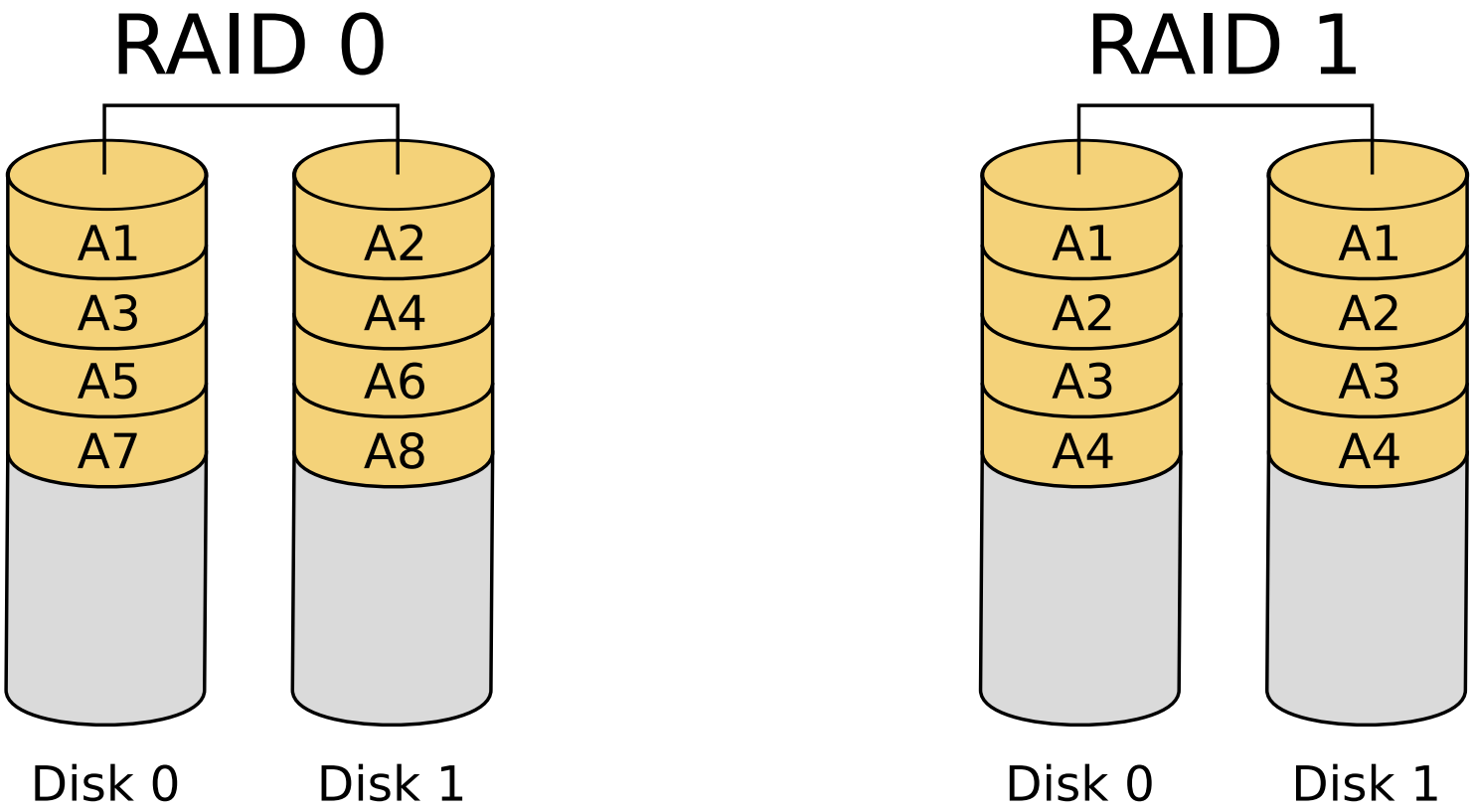
See e.g. https://en.wikipedia.org/wiki/Backup

RAID: The Idea

- Reliable, fast disks are expensive
- (Cheap) disks are unreliable, slow
- "Redundant Array of [Inexpensive;Independent] Disks"
- Use multiple (cheap) disks to get/exceed the reliability/speed of expensive disks

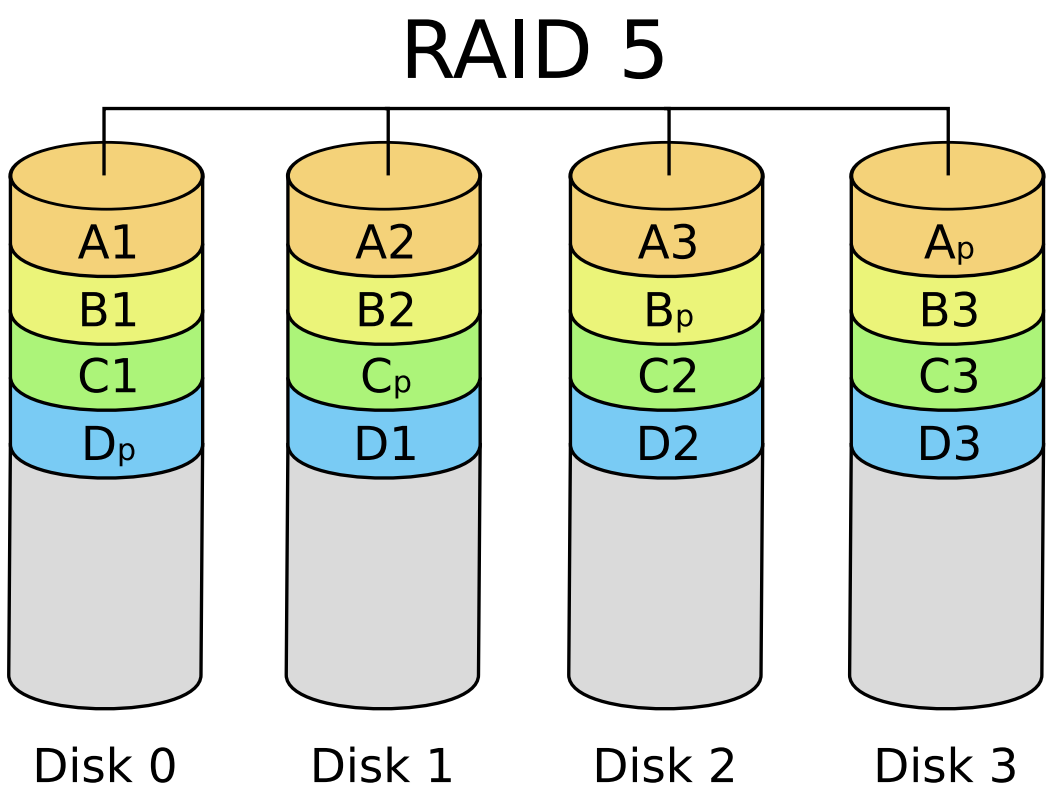
See e.g. https://en.wikipedia.org/wiki/RAID

RAID 0 + 1



en:User:Cburnett (CC BY-SA 3.0), as seen on
https://en.wikipedia.org/wiki/Standard_RAID_levels.

RAID 5



$$X_p = X_1 \oplus X_2 \oplus \ldots \oplus X_{n-1}$$

en:User:Cburnett (CC BY-SA 3.0), as seen on https://en.wikipedia.org/wiki/Standard_RAID_levels.

RAID Level Properties

Level	Disks	Avail	Failures	Read perf	Write perf
RAID 0	n	n	0	nx	nx
RAID 1	n	1	$n-1$	nx	1x
RAID 10	2x2	2	1...2	4x	2x
RAID 5	n	$n-1$	1	nx	$n-1$x
RAID 6	n	$n-2$	2	nx	$n-2$x

Performance Assumptions

- Independent links to disks
- All disks operational
- All disks identical

RAID is **never** a replacement for backup! Why?

RAID Performance

Level	Space (disks) used	Space avail	Single block read	Single block write	Stripe write	Bad block read	Bad block write
RAID 0	nx	nx	1x	1x	nx	N/A	N/A
RAID 1	nx	1x	1x	nx	nx	2x	1x
RAID 10	4x	2x	1x	2x	1x	1x	1.5x
RAID 5	nx	$n-1$x	1x	nx, 2x	nx	$n-1$x	nx

Assumptions

- Space as multiples of single disk size
- Read/write as multiple of data size
- Defunct disk known in advance
- Often, parallelism possible

Modern File Systems

- Large directory support (B-Trees etc.)
- Consistency (write ordering)
- Journaling
- Snapshots (Copy-on-Write)[2]
- Deduplication (Retroactive Copy-on-Write)

[2] Addresses some backup issues, not all