Betriebssysteme

11. Tutorium - Storage

Peter Bohner

23. Januar 2024

ITEC - Operating Systems Group

ÜB 9

- Lief gut - von meiner Seite aus kein Besprechungsbedarf
- Fragen euerseits?

Wiederholung - Scheduling

- Was ist ein Long und ein Short-Term scheduler?

Wiederholung - Scheduling

- Was ist ein Long und ein Short-Term scheduler? LTS: Was kommt in die Run-Queue, STS: Was läuft auf der CPU?

Wiederholung - Scheduling

- Was ist ein Long und ein Short-Term scheduler? LTS: Was kommt in die Run-Queue, STS: Was läuft auf der CPU?
- Unterschied mechanism - policy? Wieso wichtig?

Wiederholung - Scheduling

- Was ist ein Long und ein Short-Term scheduler? LTS: Was kommt in die Run-Queue, STS: Was läuft auf der CPU?
- Unterschied mechanism - policy? Wieso wichtig?
- Was ist thrashing?

Wiederholung - Scheduling

- Was ist ein Long und ein Short-Term scheduler? LTS: Was kommt in die Run-Queue, STS: Was läuft auf der CPU?
- Unterschied mechanism - policy? Wieso wichtig?
- Was ist thrashing?
- Welche threading Modelle kennt ihr? Vor- / Nachteile?

Wiederholung - Scheduling

- Was ist ein Long und ein Short-Term scheduler? LTS: Was kommt in die Run-Queue, STS: Was läuft auf der CPU?
- Unterschied mechanism - policy? Wieso wichtig?
- Was ist thrashing?
- Welche threading Modelle kennt ihr? Vor- / Nachteile? One-to-One (Langsam, Kernel Support, einfach), Many-to-one (kein Kernel support, kein SMP), Many-to-Many (upcalls, komplexität)
- Unterschied KLT, KMT?

Wiederholung - Klausuraufgaben

- Speicher - Segmentation: WS21 final1 T3
- Caching: WS1617 final1 T4 c,d

Device Management

Device Management

What kind of I/O Devices do you find in a typical system?

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Block Devices

- Offer random access to fixed-size blocks
- Applications typically deal with a file system on top of the device
- Examples?

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Block Devices

- Offer random access to fixed-size blocks
- Applications typically deal with a file system on top of the device
- Examples? SSD, HDD, ...

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Block Devices

- Offer random access to fixed-size blocks
- Applications typically deal with a file system on top of the device
- Examples? SSD, HDD, ...

Character Devices

- Provide a stream of characters
- Examples?

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Block Devices

- Offer random access to fixed-size blocks
- Applications typically deal with a file system on top of the device
- Examples? SSD, HDD, ...

Character Devices

- Provide a stream of characters
- Examples? Mice, Keyboard, (classic) text terminals

Communication with Devices

Port Based I/O

Communication with Devices

Port Based I/O

Separate address space with dedicated instructions for reading/writing

- \+ Clear distinction in code $\Rightarrow$ Optimizing easier (reordering, caching, ...)
- \- Less flexible, often lower performance

Communication with Devices

Port Based I/O

Separate address space with dedicated instructions for reading/writing

- \+ Clear distinction in code ⇒ Optimizing easier (reordering, caching, ...)
- \- Less flexible, often lower performance

Memory-mapped I/O

Device registers are mapped into the physical address space. How do you access that?

Communication with Devices

Port Based I/O

Separate address space with dedicated instructions for reading/writing

+ Clear distinction in code ⇒ Optimizing easier (reordering, caching, ...)
- Less flexible, often lower performance

Memory-mapped I/O

Device registers are mapped into the physical address space. How do you access that? Normal instructions!

+ Higher flexibility: Virtual memory, larger instruction set, mostly transparent
- Some special rules apply to I/O regions software needs to be aware of

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

- Devices can access the physical memory *without* involving the CPU

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

- Devices can access the physical memory *without* involving the CPU
- Needs special setup from the OS to know how and what to read/write

Memory Mapped Files

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

- Devices can access the physical memory *without* involving the CPU
- Needs special setup from the OS to know how and what to read/write

Memory Mapped Files

- OS abstraction: Treat a file like a normal range of virtual memory

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

- Devices can access the physical memory *without* involving the CPU
- Needs special setup from the OS to know how and what to read/write

Memory Mapped Files

- OS abstraction: Treat a file like a normal range of virtual memory
- No real relation to DMA, though the OS might use it to synchronize Memory Mapped Files with the underlying device

I/O Operations

How does the OS know an I/O operation is finished?

I/O Operations

How does the OS know an I/O operation is finished?

- Polling

I/O Operations

How does the OS know an I/O operation is finished?

- Polling $\Rightarrow$ Periodically check device registers

I/O Operations

How does the OS know an I/O operation is finished?

- Polling $\Rightarrow$ Periodically check device registers
- Interrupts

I/O Operations

How does the OS know an I/O operation is finished?

- Polling $\Rightarrow$ Periodically check device registers
- Interrupts $\Rightarrow$ I/O devices send an interrupt signal

Hard Disks

Hard Disk Layout

What parts can you find in a hard disk?

Hard Disk Layout

What parts can you find in a hard disk?

- Heads

Hard Disk Layout

What parts can you find in a hard disk?

- Heads
- Arms

Hard Disk Layout

What parts can you find in a hard disk?

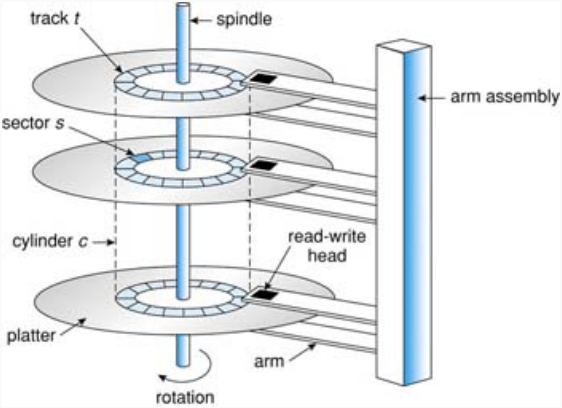
- Heads
- Arms
- Platters

Hard Disk Layout

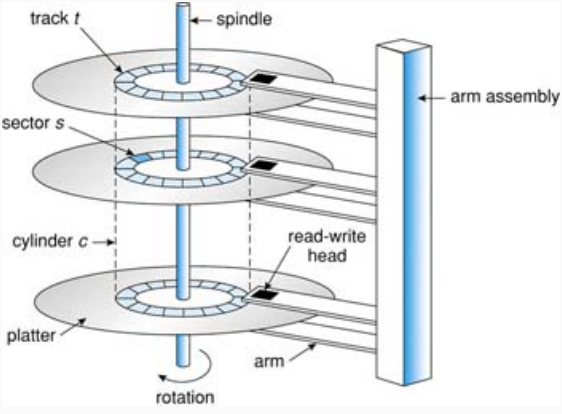
What parts can you find in a hard disk?

- Heads
- Arms
- Platters
- Spindle

Hard Disk Layout

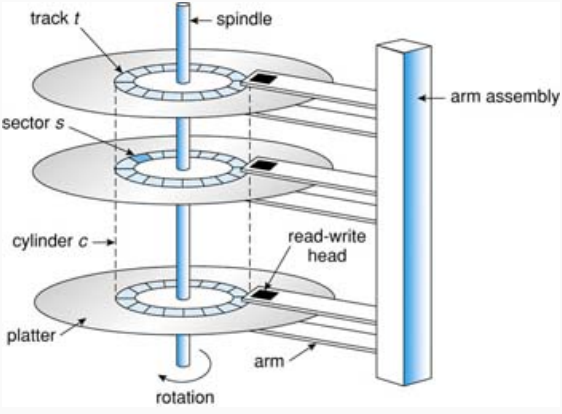


Hard Disk Layout



What do they do?

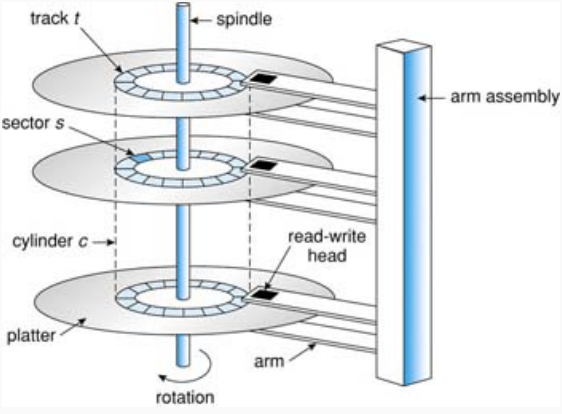
Hard Disk Layout



What do they do?

- Spindle: Spin connected platters!

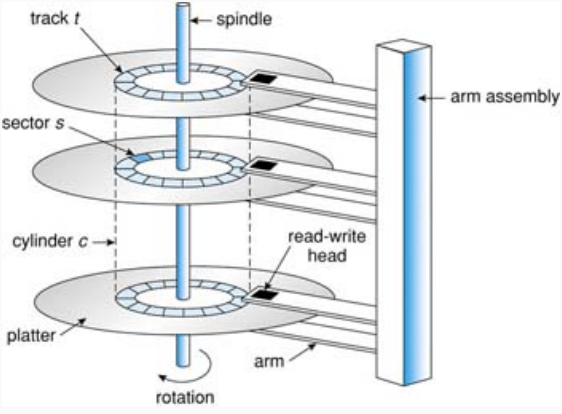
Hard Disk Layout



What do they do?

- Spindle: Spin connected platters!
- Head: Read/Write

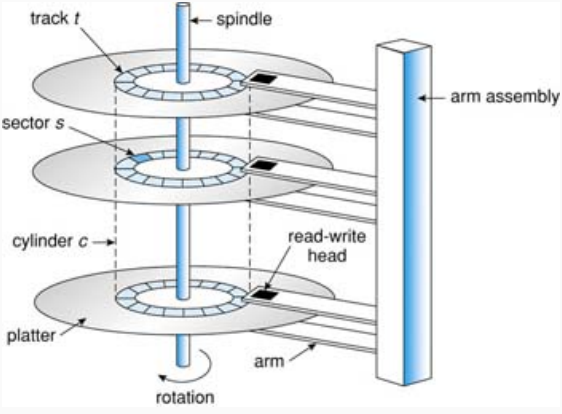
Hard Disk Layout



What do they do?

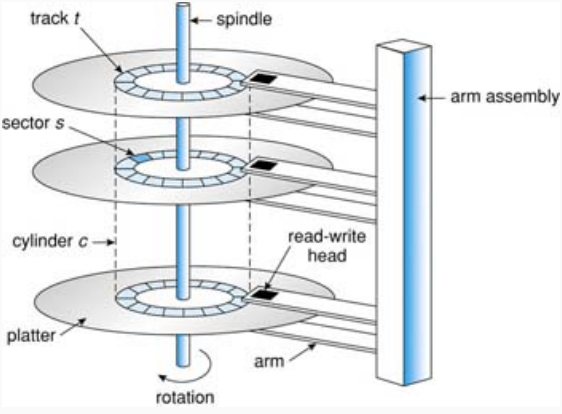
- Spindle: Spin connected platters!
- Head: Read/Write
- Arm: Move heads

Hard Disk Layout



How can you address data (512 byte blocks typically) on the disk?

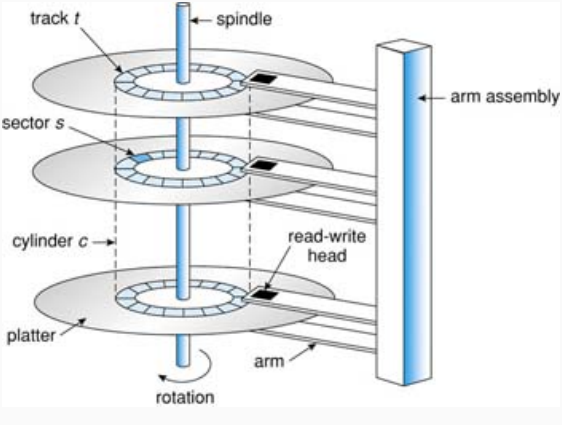
Hard Disk Layout



How can you address data (512 byte blocks typically) on the disk?

- Cylinder - Head - Sector (CHS). Limited to „small“ disks (< 8GB), rarely used these days

Hard Disk Layout



How can you address data (512 byte blocks typically) on the disk?

- Cylinder - Head - Sector (CHS). Limited to „small“ disks (< 8GB), rarely used these days
- Logical Block Addressing (LBA). Each data block has its own unique number.

How could you optimize the OS $\Leftrightarrow$ Disk interface?

Native-Command-Queuing. OS sends reads and writes in batches and (the disk | the OS) reorders them based on internal geometry.

How could you optimize the OS $\Leftrightarrow$ Disk interface?

Native-Command-Queuing. OS sends reads and writes in batches and *the disk* reorders them based on internal geometry.

Wear and Tear

What do you do when a sector is damaged?

Wear and Tear

What do you do when a sector is damaged?

Disk marks it as such and never uses it again ⇒ *Sector sparing*.

What adverse effect might this have?

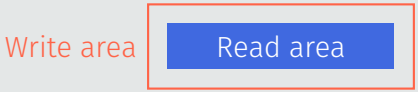
Wear and Tear

What do you do when a sector is damaged?

Disk marks it as such and never uses it again $\Rightarrow$ *Sector sparing*.
What adverse effect might this have? OS disk scheduler is unaware and optimizes for wrong geometry.

Hard Disk Layout - More Data!

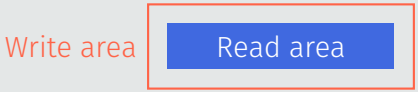
Optimizing storage



What can you do with this?

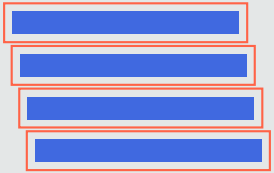
Hard Disk Layout - More Data!

Optimizing storage



What can you do with this?

Conventional layout

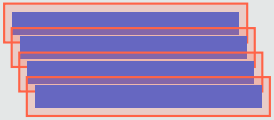


Hard Disk Layout - More Data!

Shingled Magnet Recording

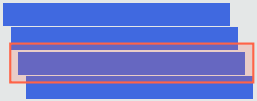
Hard Disk Layout - More Data!

Shingled Magnet Recording



Hard Disk Layout - More Data!

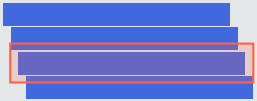
Shingled Magnet Recording



What happens when you write to this track?

Hard Disk Layout - More Data!

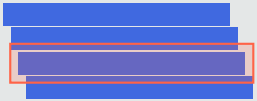
Shingled Magnet Recording



What happens when you write to this track? You overwrite the adjacent ones!

Hard Disk Layout - More Data!

Shingled Magnet Recording

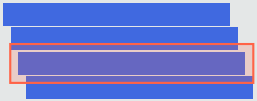


What happens when you write to this track? You overwrite the adjacent ones!

⇒ Append only and group shingled tracks

Hard Disk Layout - More Data!

Shingled Magnet Recording



What happens when you write to this track? You overwrite the adjacent ones!

⇒ Append only and group shingled tracks

⇒ Can rewrite the whole group at once

Hard Disk Layout - Shingled Magnet Recording

How can such a device interface with the OS?

Hard Disk Layout - Shingled Magnet Recording

How can such a device interface with the OS?

- Pretend you are a normal disk. Buffer writes in a normal zone and flush them once they fill up a group.
  $\Rightarrow$ *Device Managed*

Hard Disk Layout - Shingled Magnet Recording

How can such a device interface with the OS?

- Pretend you are a normal disk. Buffer writes in a normal zone and flush them once they fill up a group.
  $\Rightarrow$ *Device Managed*
- Tell the OS where your shingled zones are. The OS needs to write carefully to not destroy data
  $\Rightarrow$ *Host Managed*

Hard Disk Layout - Shingled Magnet Recording

How can such a device interface with the OS?

- Pretend you are a normal disk. Buffer writes in a normal zone and flush them once they fill up a group.
  $\Rightarrow$ *Device Managed*
- Tell the OS where your shingled zones are. The OS needs to write carefully to not destroy data
  $\Rightarrow$ *Host Managed*
- Compromise. Tell the OS where your singled zones are and expose their tail. If the OS writes to the tail, directly commit it - else buffer.
  $\Rightarrow$ *Host Aware*

Solid-State Drives

NAND based flash memory

Rejoice, TI might be useful once

How long do writes/reads take normally?

NAND based flash memory

Rejoice, TI might be useful once

How long do writes/reads take normally?

- Reading a page: 25µs

Rejoice, TI might be useful once

How long do writes/reads take normally?

- Reading a page: 25µs
- Writing a page: 250µs

NAND based flash memory

Rejoice, TI might be useful once

How long do writes/reads take normally?

- Reading a page: 25µs
- Writing a page: 250µs
- Erasing a block:

NAND based flash memory

Rejoice, TI might be useful once

How long do writes/reads take normally?

- Reading a page: 25µs
- Writing a page: 250µs
- Erasing a block: 2ms

What happens when you just write to a random page?

Rejoice, TI might be useful once

How long do writes/reads take normally?

- Reading a page: 25µs
- Writing a page: 250µs
- Erasing a block: 2ms

What happens when you just write to a random page?

Speeding things up

What could you change so writing pages is faster?

NAND based flash memory

Rejoice, TI might be useful once

How long do writes/reads take normally?

- Reading a page: 25µs
- Writing a page: 250µs
- Erasing a block: 2ms

What happens when you just write to a random page?

Speeding things up

What could you change so writing pages is faster?

- Keep around spare *erased* pages

$\Rightarrow$ You do not pay the erase penalty!

- When do you create / reserve / erase those spare pages?

Rejoice, TI might be useful once

How long do writes/reads take normally?

- Reading a page: 25µs
- Writing a page: 250µs
- Erasing a block: 2ms

What happens when you just write to a random page?

Speeding things up

What could you change so writing pages is faster?

- Keep around spare *erased* pages

$\Rightarrow$ You do not pay the erase penalty!

- When do you create / reserve / erase those spare pages? Probably in the background. Any problems?

Rejoice, TI might be useful once

How long do writes/reads take normally?

- Reading a page: 25µs
- Writing a page: 250µs
- Erasing a block: 2ms

What happens when you just write to a random page?

Speeding things up

What could you change so writing pages is faster?

- Keep around spare *erased* pages

$\Rightarrow$ You do not pay the erase penalty!

- When do you create / reserve / erase those spare pages? Probably in the background. Any problems? Might get exhausted if you write too much data in a short timeframe or the disk is full!

Deleting files

What happens when you delete a file? What effect does that have on the SSD performance?

Deleting files

What happens when you delete a file? What effect does that have on the SSD performance? The block is not freed ⇒ Can't be used as an erased empty page

What can the OS do to combat that?

Deleting files

What happens when you delete a file? What effect does that have on the SSD performance? The block is not freed ⇒ Can't be used as an erased empty page

What can the OS do to combat that?

The `trim` command

Can be issued by the OS to tell the SSD firmware what pages can be safely erased.

RAID

RAID

What is that?

A **R**edundant **A**rray of **I**ndependent/**I**nexpensive **D**isks

RAID

What is that?

A **R**edundant **A**rray of **I**ndependent/**I**nexpensive **D**isks

Why would you use that?

RAID

What is that?

A **R**edundant **A**rray of **I**ndependent/**I**nexpensive **D**isks

Why would you use that?

- Probably cheaper than a SLED (Single Large Expensive Disk)

RAID

What is that?

A **R**edundant **A**rray of **I**ndependent/**I**nexpensive **D**isks

Why would you use that?

- Probably cheaper than a SLED (Single Large Expensive Disk)
- Might be more resilient

RAID

What is that?

A **R**edundant **A**rray of **I**ndependent/**I**nexpensive **D**isks

Why would you use that?

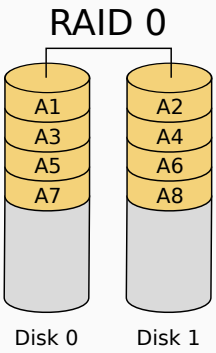
- Probably cheaper than a SLED (Single Large Expensive Disk)
- Might be more resilient
- Might be faster

RAID - And now?

Great, you now have multiple disks. How do you store your files on them?

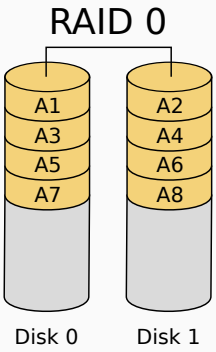
- „I like to live dangerously“ - RAID Level 0
- Mirroring: RAID Level 1
- Historic variants: RAID Level 2 and 3
- Block striping and parity: RAID Level 4
- Block striping and *distributed* parity: RAID Level 5

RAID 0



Benefits / Drawbacks?

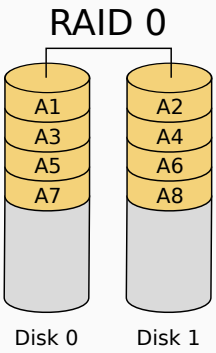
RAID 0



Benefits / Drawbacks?

+ Extremely fast (parallel reads and writes)

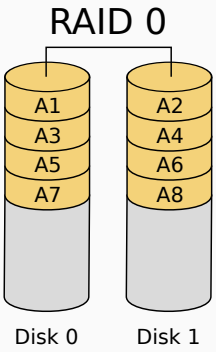
RAID 0



Benefits / Drawbacks?

+ Extremely fast (parallel reads and writes)
+ Can use full capacity

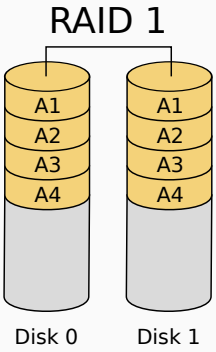
RAID 0



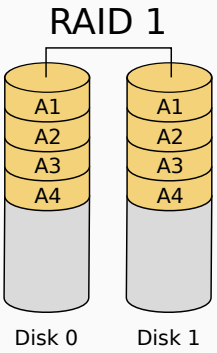
Benefits / Drawbacks?

+ Extremely fast (parallel reads and writes)
+ Can use full capacity
- If a single disk fails your files are toast

RAID 1



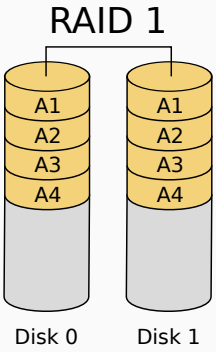
Benefits / Drawbacks?



Benefits / Drawbacks?

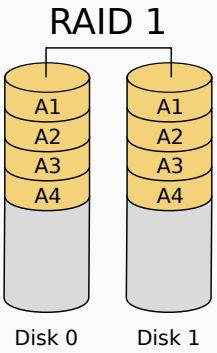
+ You can lose all but one disk without losing data

RAID 1



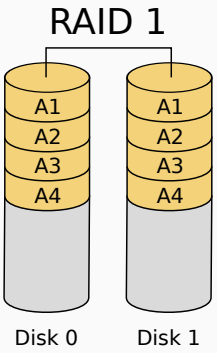
Benefits / Drawbacks?

+ You can lose all but one disk without losing data
+ Parallel reads possible



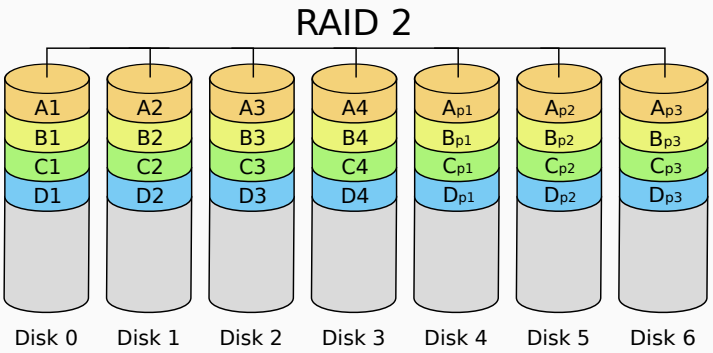
Benefits / Drawbacks?

+ You can lose all but one disk without losing data
+ Parallel reads possible
- Writes slower as they need to write to all disks



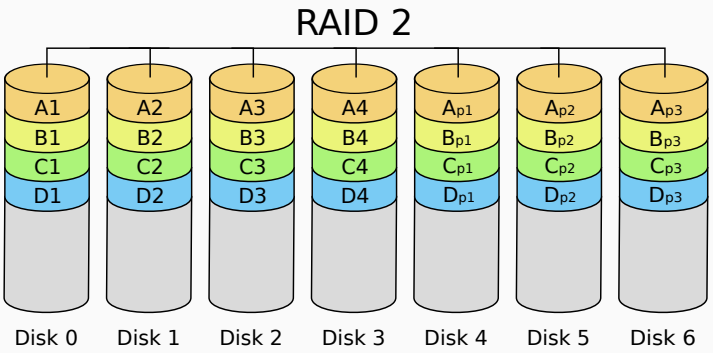
Benefits / Drawbacks?

+ You can lose all but one disk without losing data
+ Parallel reads possible
- Writes slower as they need to write to all disks
- Size equals the size of a single disk



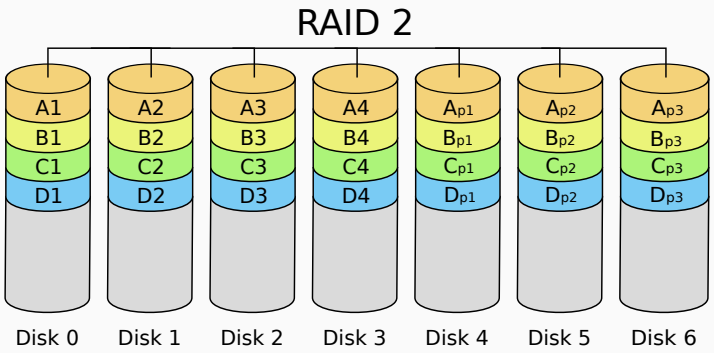
What is that?

- Have $\log_2(N)$ parity disk



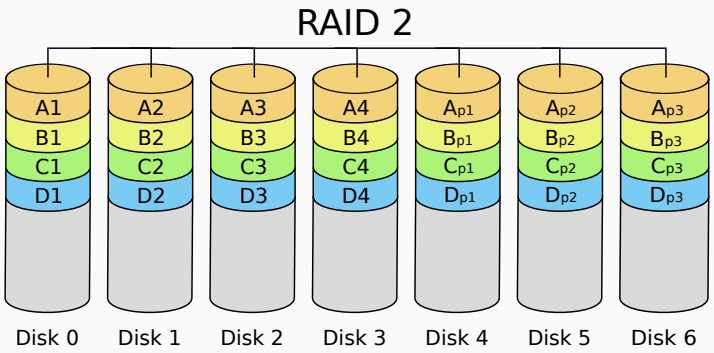
What is that?

- Have $\log_2(N)$ parity disk
- Stripe data at the *bit* level



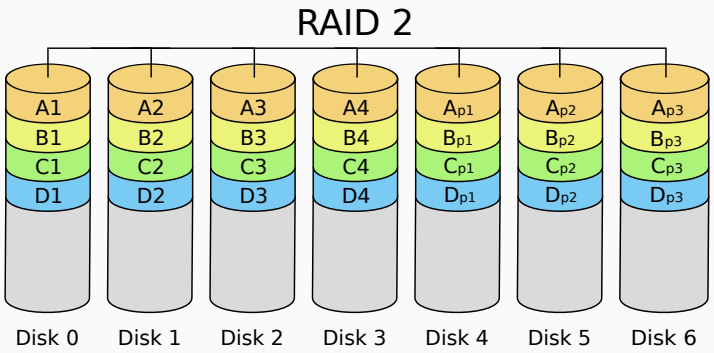
What is that?

- Have $\log_2(N)$ parity disk
- Stripe data at the *bit* level
- Use a hamming code of proper size

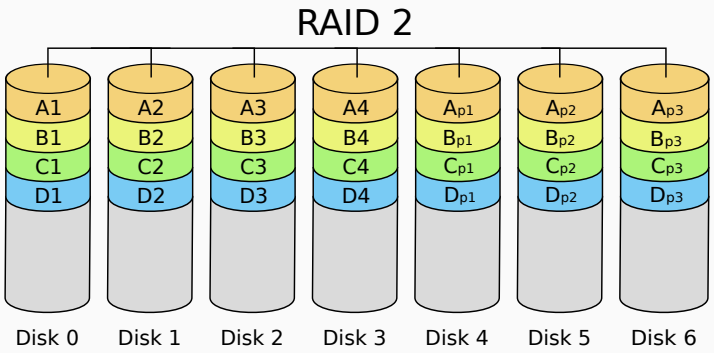


What is that?

- Have $\log_2(N)$ parity disk
- Stripe data at the *bit* level
- Use a hamming code of proper size
- Spin the disks in lockstep (so you read all bits of your word at once)

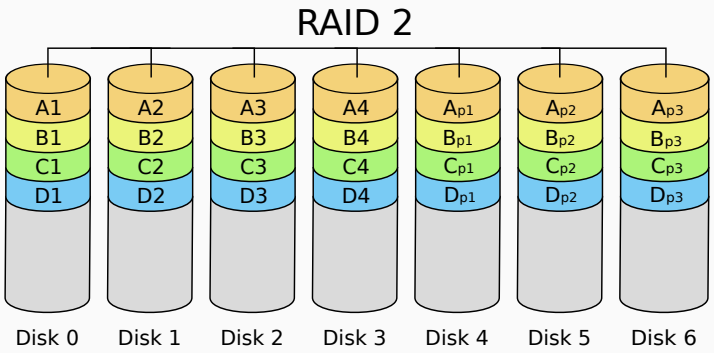


Benefits / Drawbacks



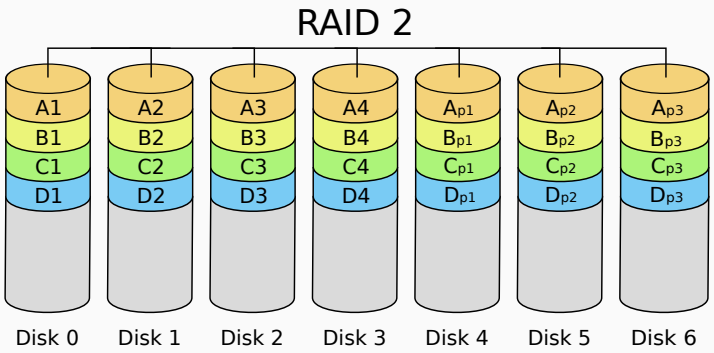
Benefits / Drawbacks

+ External error checking



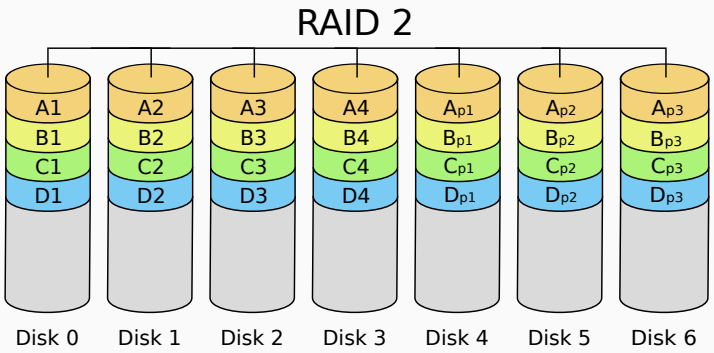
Benefits / Drawbacks

+ External error checking
- Really slow



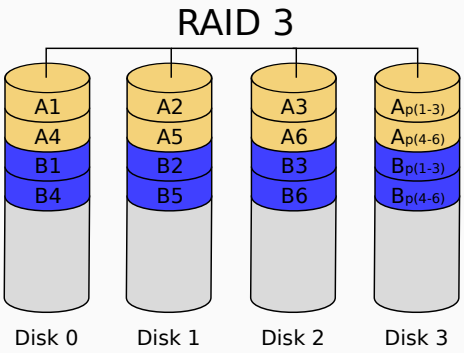
Benefits / Drawbacks

+ External error checking
- Really slow
- Not that useful as disks have internal error checking by now



Benefits / Drawbacks

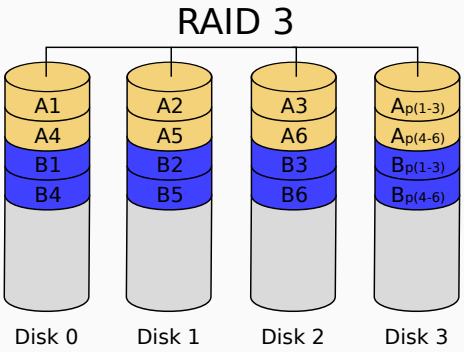
+ External error checking
- Really slow
- Not that useful as disks have internal error checking by now
- Spins in lockstep ⇒ Can only service one request at a time



What is that?

- Have a dedicated parity disk

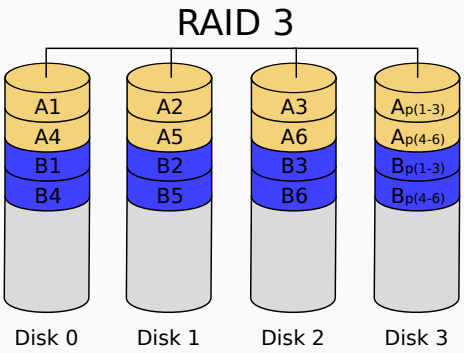
RAID 3



What is that?

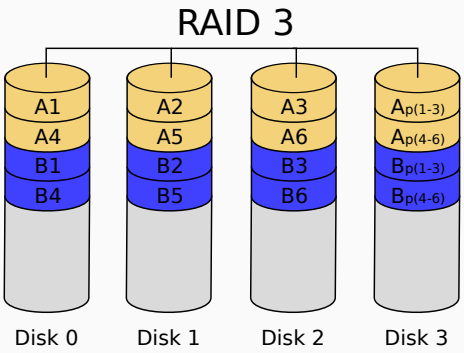
- Have a dedicated parity disk
- Stripe data at the *byte* level

RAID 3

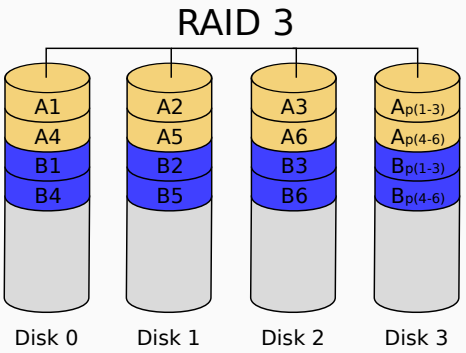


What is that?

- Have a dedicated parity disk
- Stripe data at the *byte* level
- Spin the disks in lockstep (so you read all bytes of your word at once)

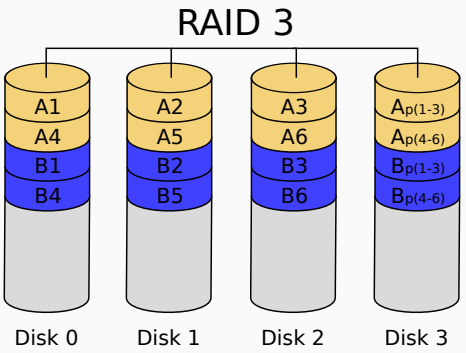


Benefits / Drawbacks



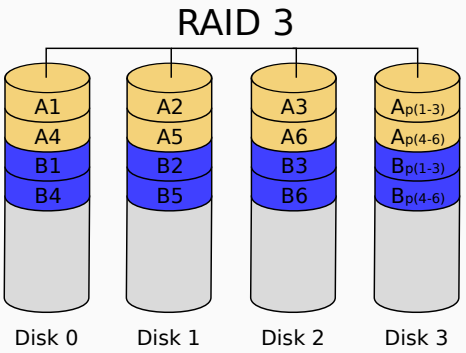
Benefits / Drawbacks

+ You can lose a disk and restore it using the parity



Benefits / Drawbacks

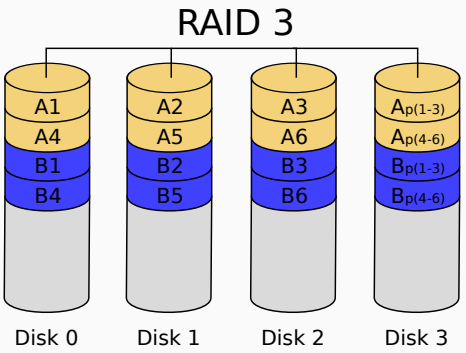
+ You can lose a disk and restore it using the parity
- Slow when reading/writing small files at random locations



Benefits / Drawbacks

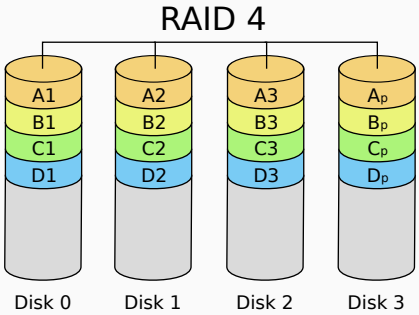
+ You can lose a disk and restore it using the parity
- Slow when reading/writing small files at random locations
- Spins in lockstep ⇒ Can only service one request at a time

RAID 3



Benefits / Drawbacks

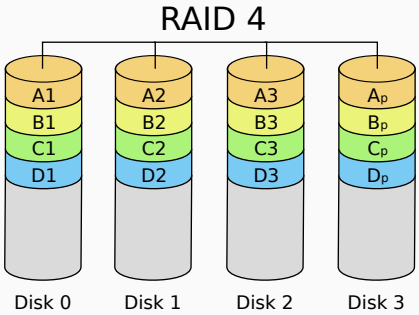
+ You can lose a disk and restore it using the parity
- Slow when reading/writing small files at random locations
- Spins in lockstep ⇒ Can only service one request at a time
- Every write and read hits the same single parity disk



What is that?

- Have a dedicated parity disk

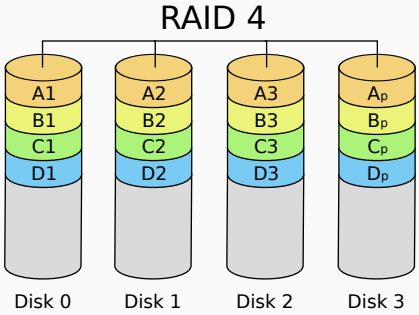
RAID 4



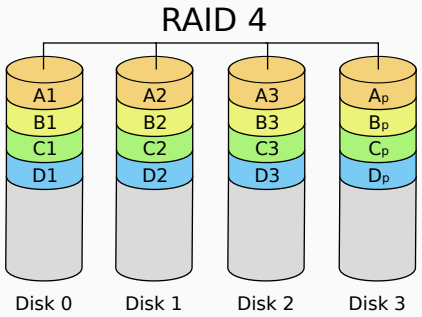
What is that?

- Have a dedicated parity disk
- Stripe data at the *block* level

RAID 4

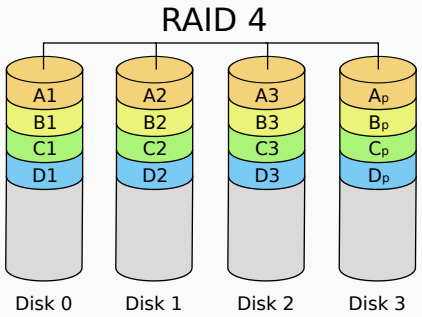


Benefits / Drawbacks



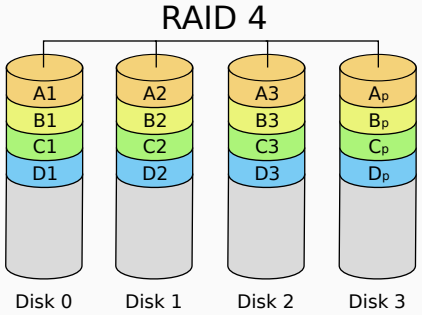
Benefits / Drawbacks

+ You can lose a disk and restore it using the parity



Benefits / Drawbacks

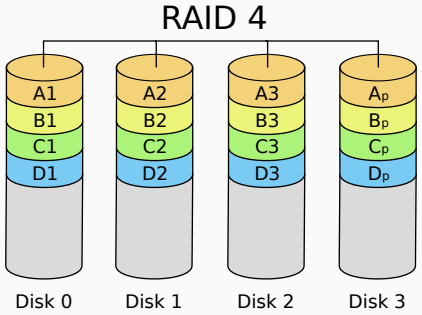
+ You can lose a disk and restore it using the parity
+ Good read performance



Benefits / Drawbacks

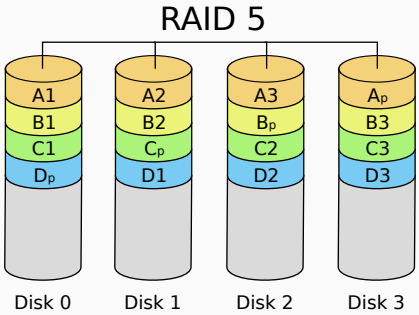
+ You can lose a disk and restore it using the parity
+ Good read performance
- Every write and read hits the same single parity disk ⇒ Bottleneck, prone to failure

RAID 4



Benefits / Drawbacks

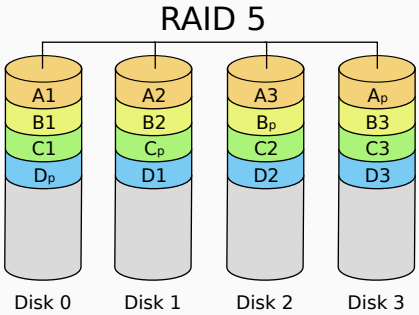
+ You can lose a disk and restore it using the parity
+ Good read performance
- Every write and read hits the same single parity disk $\Rightarrow$ Bottleneck, prone to failure
- Slow writes (write to same parity disk)



What is that?

- Stripe data at the *block* level

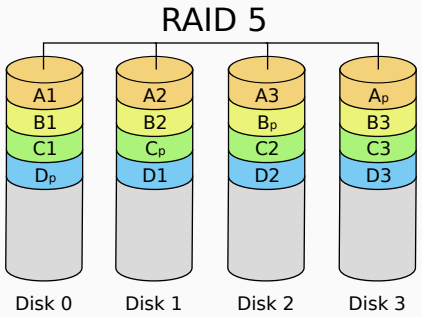
RAID 5



What is that?

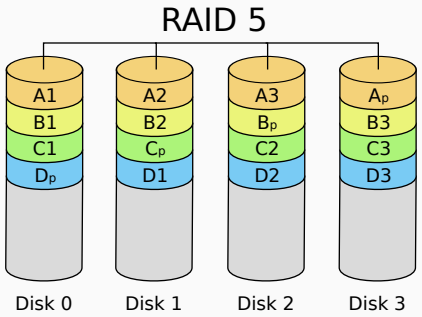
- Stripe data at the *block* level
- Distribute parity across your disks

RAID 5



Benefits / Drawbacks

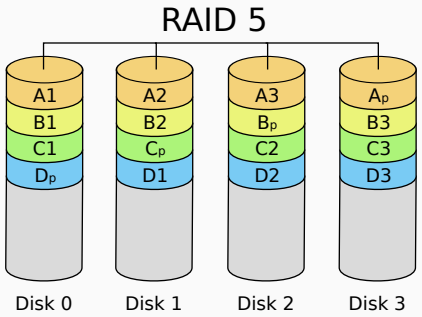
RAID 5



Benefits / Drawbacks

+ You can lose a disk and restore it using the parity

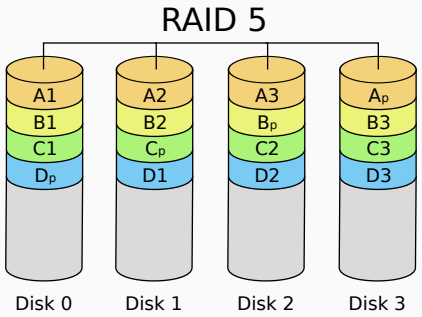
RAID 5



Benefits / Drawbacks

+ You can lose a disk and restore it using the parity
+ Good read performance

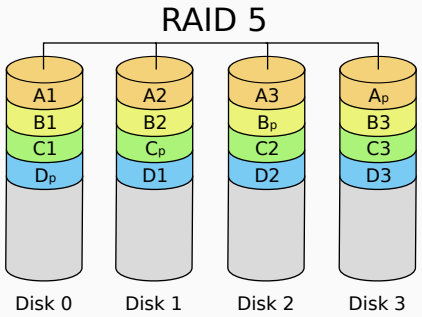
RAID 5



Benefits / Drawbacks

+ You can lose a disk and restore it using the parity
+ Good read performance
+ Okay write performance

RAID 5



Benefits / Drawbacks

+ You can lose a disk and restore it using the parity
+ Good read performance
+ Okay write performance
- Still slower than RAID 0 or a SLED

Compare SLED and RAID (Level 0, 1, 4, 5)

Each RAID uses 4 disks for actual data storage.

How many disks do you need?

- SLED: **1**

Compare SLED and RAID (Level 0, 1, 4, 5)

Each RAID uses 4 disks for actual data storage.

How many disks do you need?

- SLED: **1**
- RAID 0: **4**

Compare SLED and RAID (Level 0, 1, 4, 5)

Each RAID uses 4 disks for actual data storage.

How many disks do you need?

- SLED: **1**
- RAID 0: **4**
- RAID 1: **8**

Compare SLED and RAID (Level 0, 1, 4, 5)

Each RAID uses 4 disks for actual data storage.

How many disks do you need?

- SLED: **1**
- RAID 0: **4**
- RAID 1: **8**
- RAID 4: **5**

Compare SLED and RAID (Level 0, 1, 4, 5)

Each RAID uses 4 disks for actual data storage.

How many disks do you need?

- SLED: **1**
- RAID 0: **4**
- RAID 1: **8**
- RAID 4: **5**
- RAID 5: **5**

You want to modify one byte of data. How many blocks do you need to read/write?

- SLED: 1 read + 1 write

You want to modify one byte of data. How many blocks do you need to read/write?

- SLED: 1 read + 1 write
- RAID 0: 1 read + 1 write

You want to modify one byte of data. How many blocks do you need to read/write?

- SLED: 1 read + 1 write
- RAID 0: 1 read + 1 write
- RAID 1: 1 read + 2 write (1 data + 1 mirror)

You want to modify one byte of data. How many blocks do you need to read/write?

- SLED: 1 read + 1 write
- RAID 0: 1 read + 1 write
- RAID 1: 1 read + 2 write (1 data + 1 mirror)
- RAID 4: 2 read (data + old parity) + 2 write (data + new parity)

You want to modify one byte of data. How many blocks do you need to read/write?

- SLED: 1 read + 1 write
- RAID 0: 1 read + 1 write
- RAID 1: 1 read + 2 write (1 data + 1 mirror)
- RAID 4: 2 read (data + old parity) + 2 write (data + new parity)
- RAID 5: 2 read (data + old parity) + 2 write (data + new parity)

RAID vs. Backup

You are using RAID

- You accidentally delete a file.

RAID vs. Backup

You are using RAID

- You accidentally delete a file. ***GONE***
- You accidentally overwrite a file.

RAID vs. Backup

You are using RAID

- You accidentally delete a file. ***GONE***
- You accidentally overwrite a file. ***GONE***
- Some data gets corrupted on one disk.

RAID vs. Backup

You are using RAID

- You accidentally delete a file. ***GONE***
- You accidentally overwrite a file. ***GONE***
- Some data gets corrupted on one disk. ***GONE*** *(probably)*
- The poor intern connects to the production database. (Or here)

RAID vs. Backup

You are using RAID

- You accidentally delete a file. ***GONE***
- You accidentally overwrite a file. ***GONE***
- Some data gets corrupted on one disk. ***GONE*** *(probably)*
- The poor intern connects to the production database. (Or here) ***GONE***
- A crypto-locker takes out your computer.

RAID vs. Backup

You are using RAID

- You accidentally delete a file. ***GONE***
- You accidentally overwrite a file. ***GONE***
- Some data gets corrupted on one disk. ***GONE*** *(probably)*
- The poor intern connects to the production database. (Or here) ***GONE***
- A crypto-locker takes out your computer. Believe it or not, ~~***JAIL***~~ ***GONE***

RAID vs. Backup

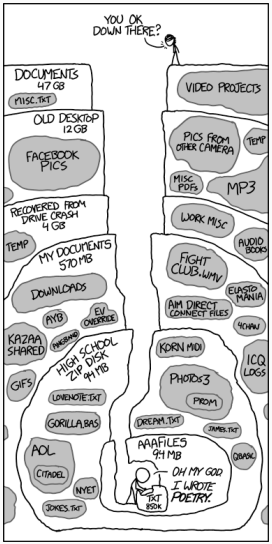
You are using RAID

- You accidentally delete a file. ***GONE***
- You accidentally overwrite a file. ***GONE***
- Some data gets corrupted on one disk. ***GONE*** *(probably)*
- The poor intern connects to the production database. (Or here) ***GONE***
- A crypto-locker takes out your computer. Believe it or not, ~~***JAIL***~~ ***GONE***

So what do we learn?

RAID IS NO SUBSTITUTE FOR A BACKUP

Ende



XKCD 1360 - Old Files

F R A G E N?

https://forms.gle/9CwJSKidKibubran9

Bis nächste Woche