



Computer Systems

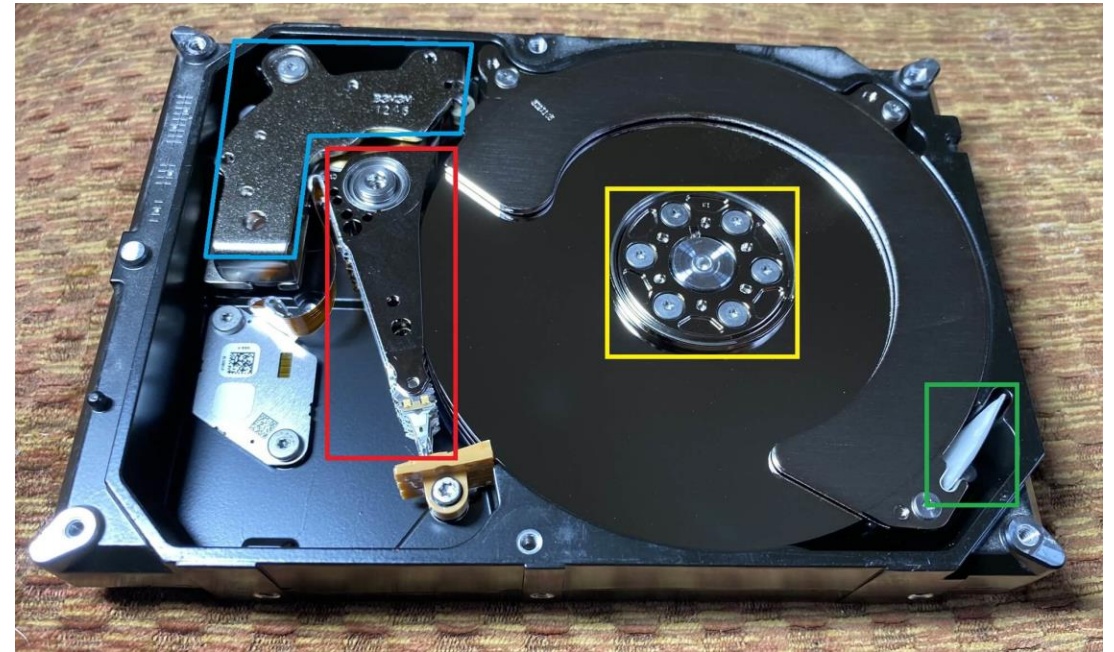
4 Bestandsystemen en File management

Files

- Computer heeft duizenden files.
- Moeten ergens op bewaard worden
 - HDD
 - SSD
 - USB
 - ...

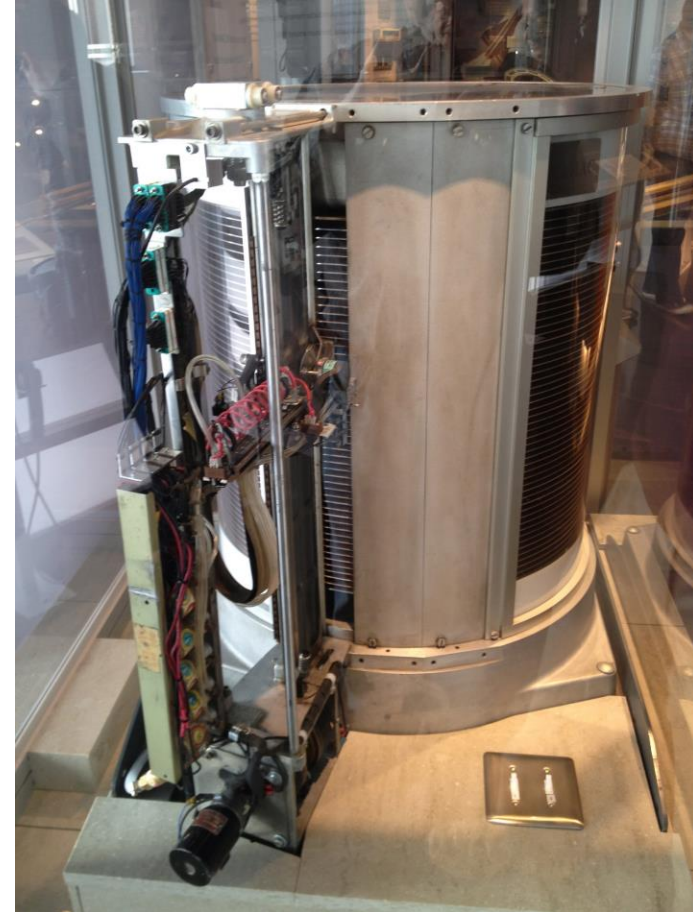


HDD: Magneetschijven



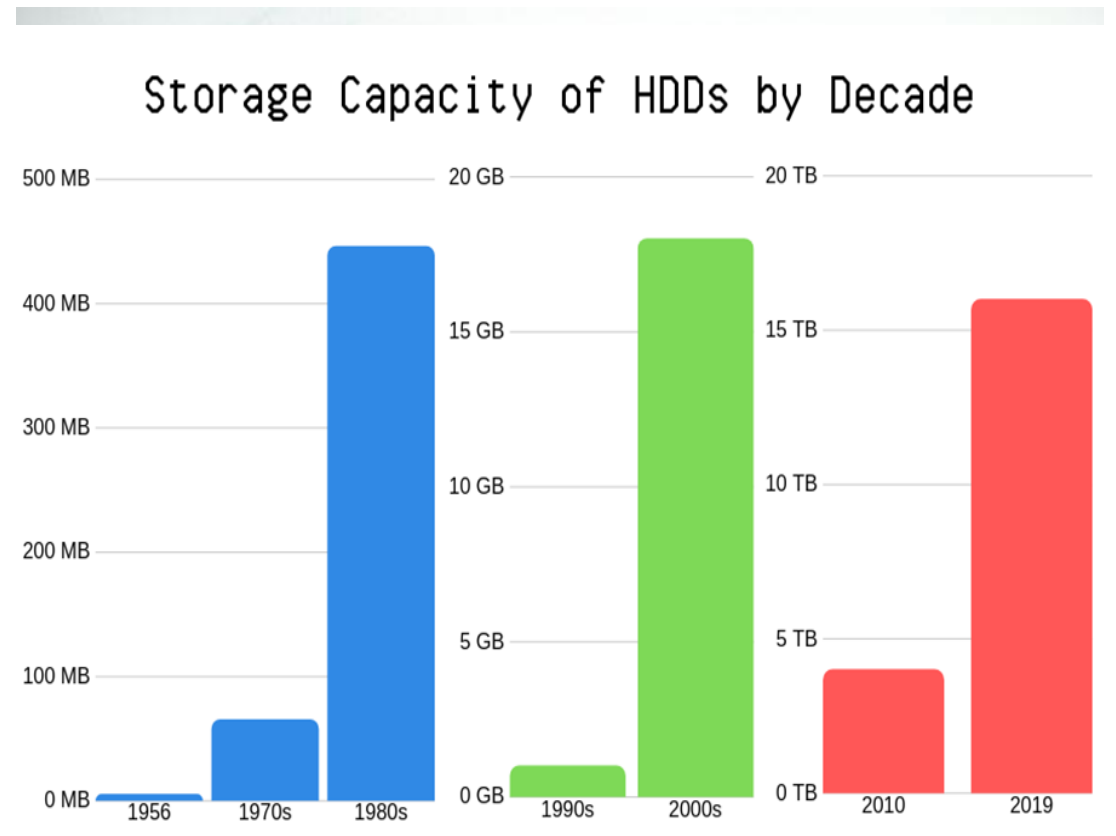
History

- Eerste commerciële schijf
 - 1956
 - IBM 350
 - 3,75MB
 - 152cm l x 74cm b x 172 cm h
 - Onderdeel van de 305 RAMAC computer
 - Ruimte van 9m x 15m
 - Leasing: \$3200/maand (\$30 874 in 2021)



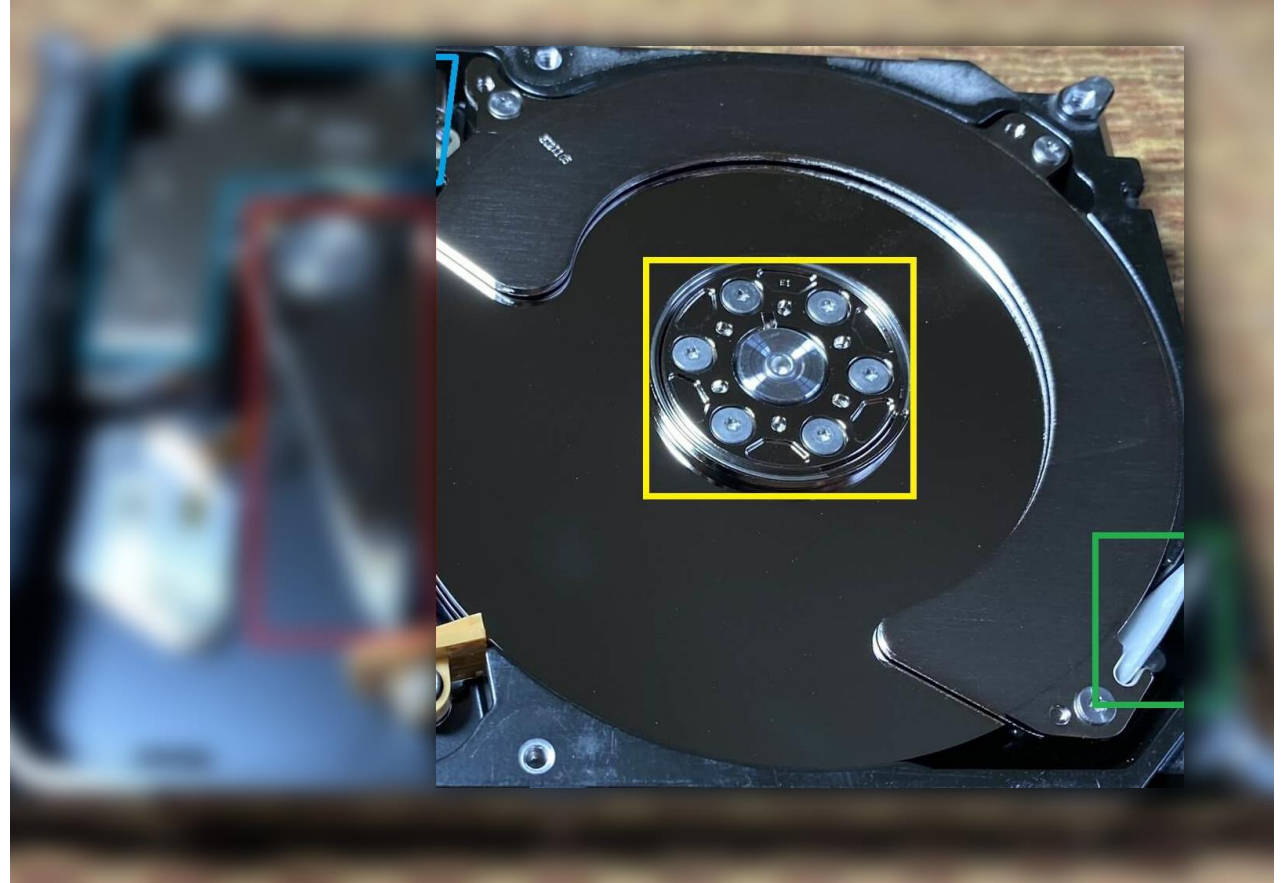
Evolutie

- Onderdelen:
 - Steeds kleiner geworden
 - Basis componenten niet veel veranderd
 - Capaciteit enorm gestegen
 - 3,75 MB in 1956
 - 20 TB in 2022
 - 50 TB tegen 2026 (Seagate)
 - 120+ TB tegen 2030 (Seagate)



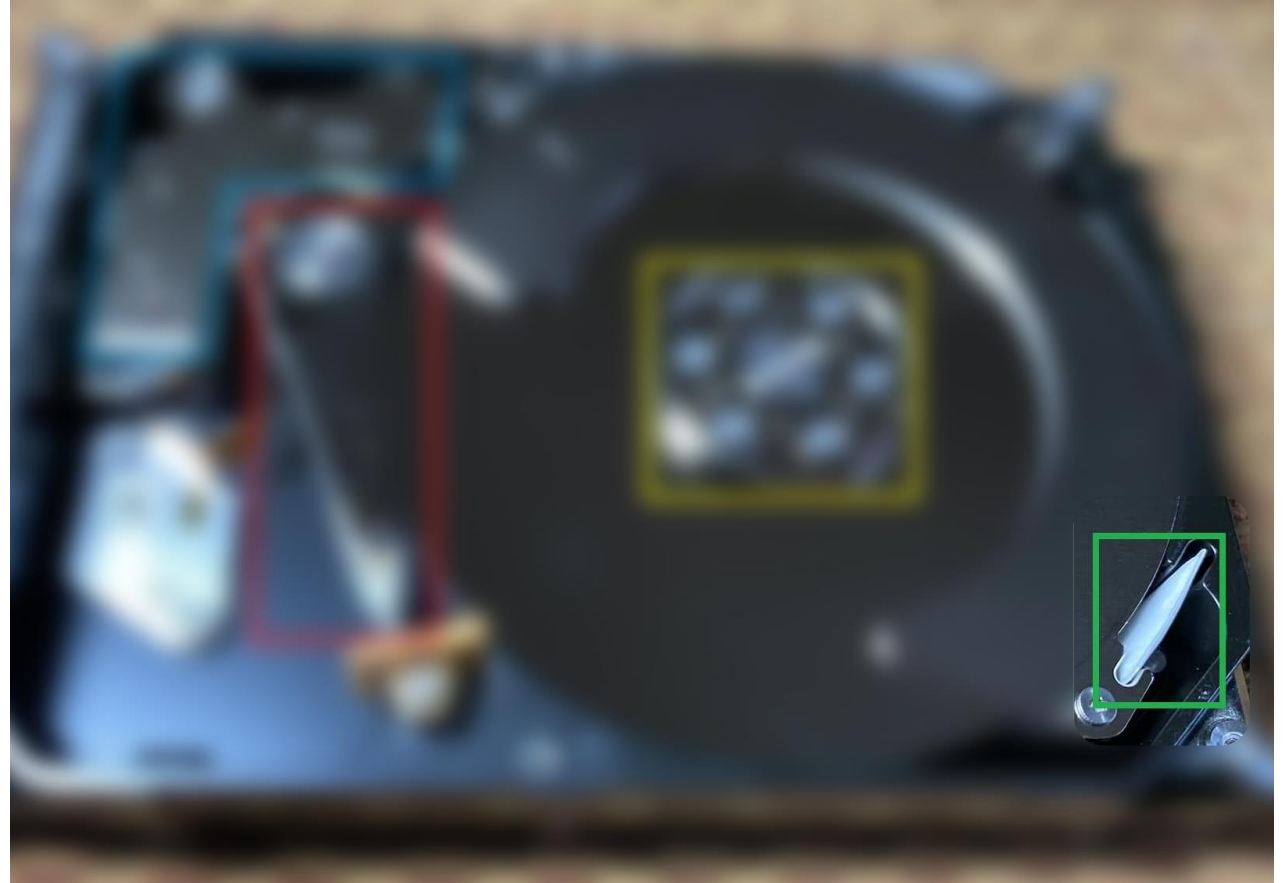
Magneetschijven

- Niet gemagnetiseerd materiaal
 - Aluminium
 - Glas (standaard)
- Beide kanten van schijf krijgen magnetische laag



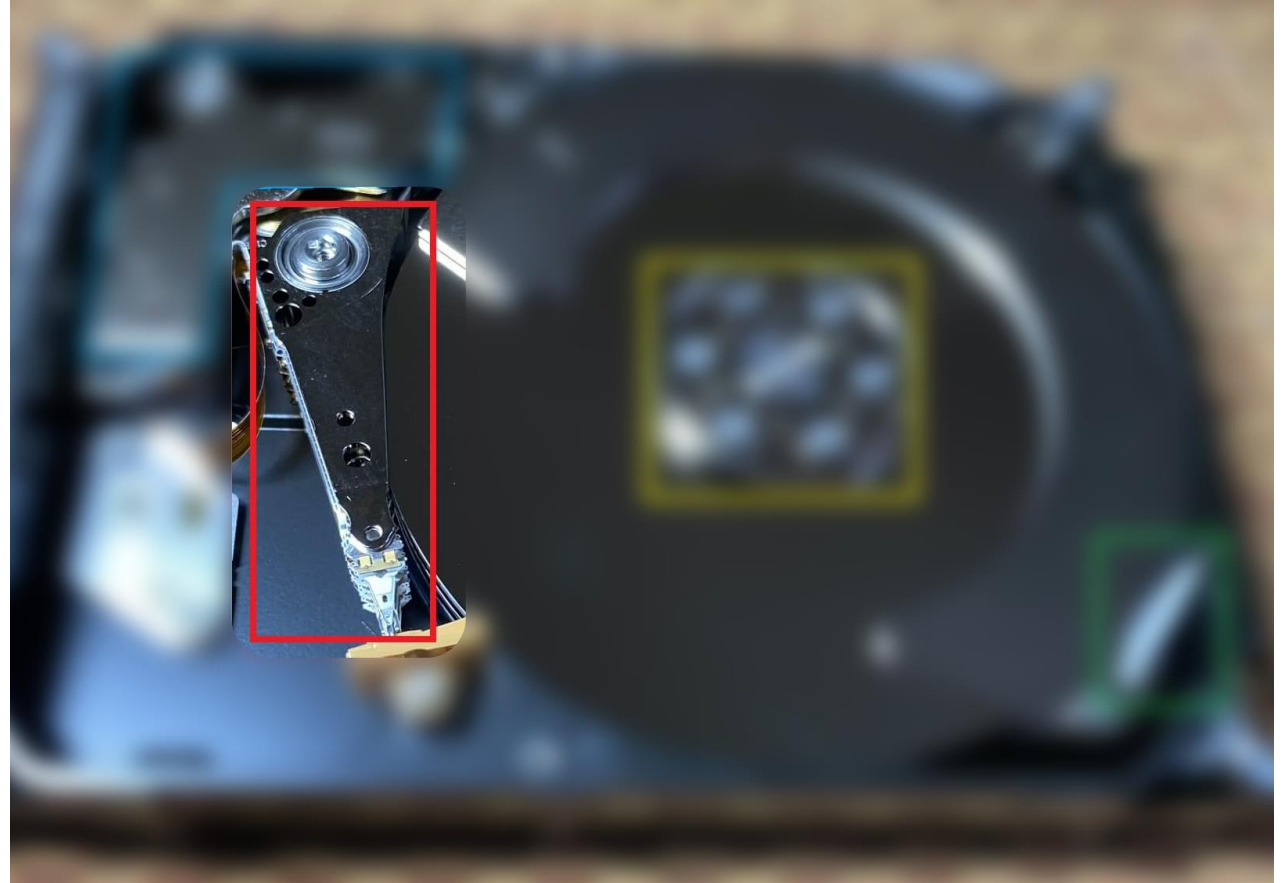
Filter

- Noodzaak:
 - Luchtcirculatie laat stof bewegen
 - HDD clean geproduceerd, toch steeds stofdeeltjes
- Filter vangt stofdeeltjes op en houdt ze vast



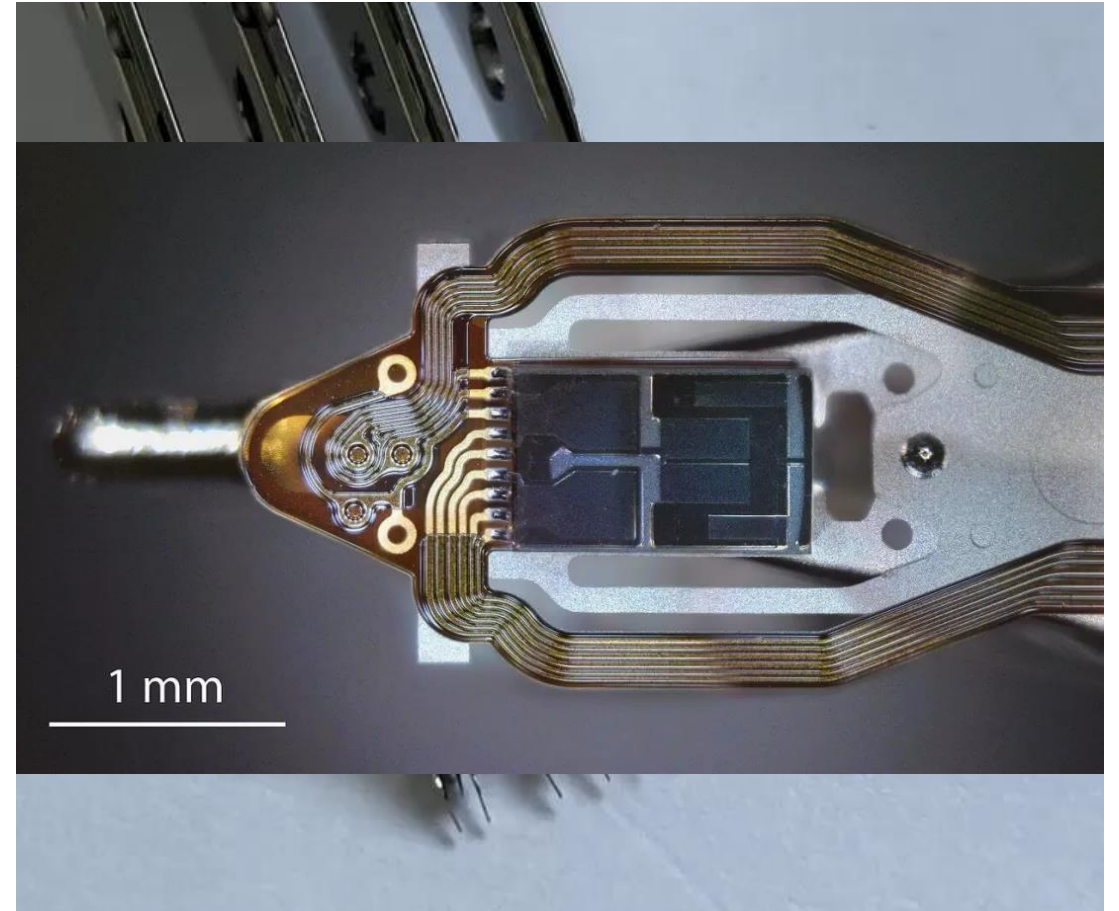
Arm

- Plaatst kop op juiste plek
- Beperkte radius:
 - Straal van HDD moet bereikt kunnen worden.



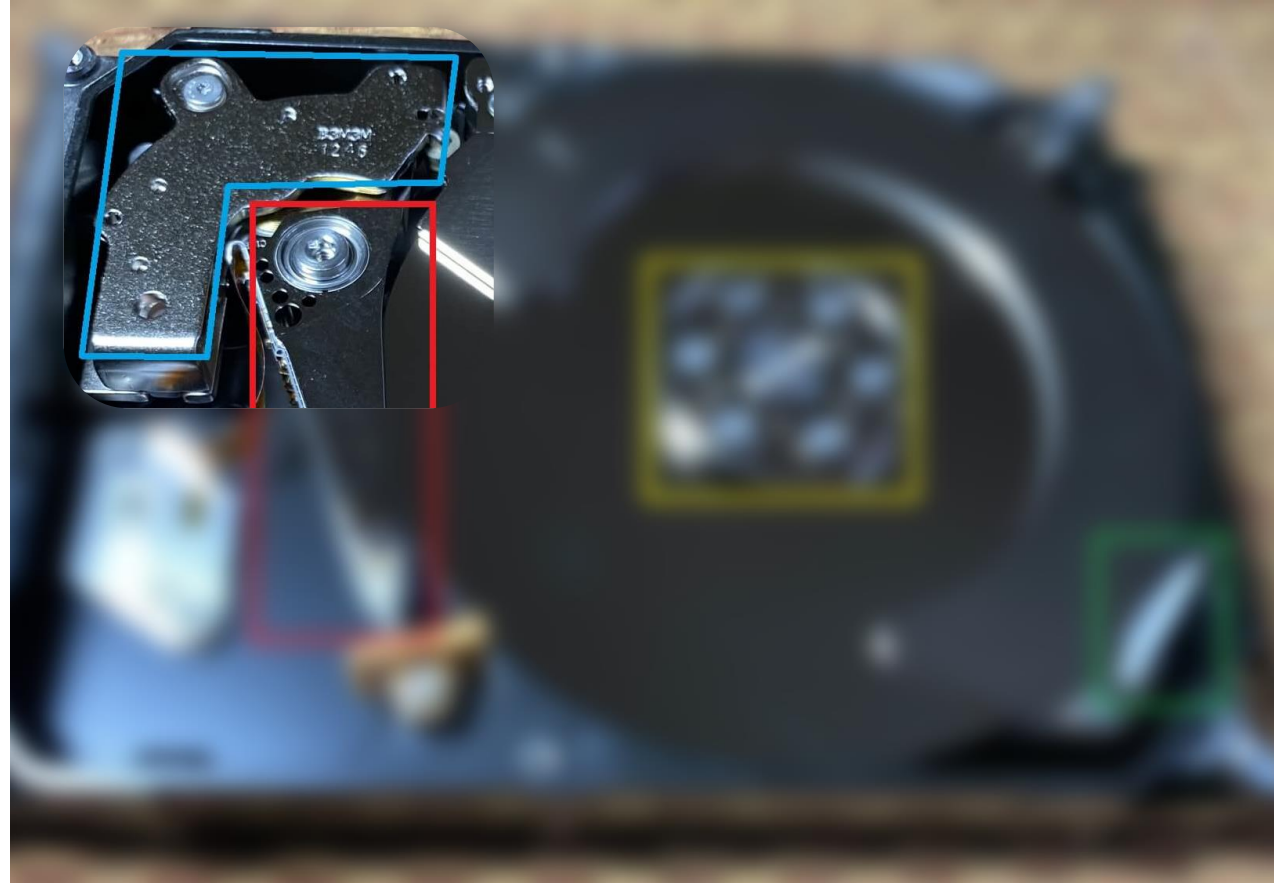
Lees- en schrijfkop

- Beweegt over mangeetschijven
- 1 kop voor zowel lezen als schrijven
- Meerdere koppen:
 - Per laag 1 kop



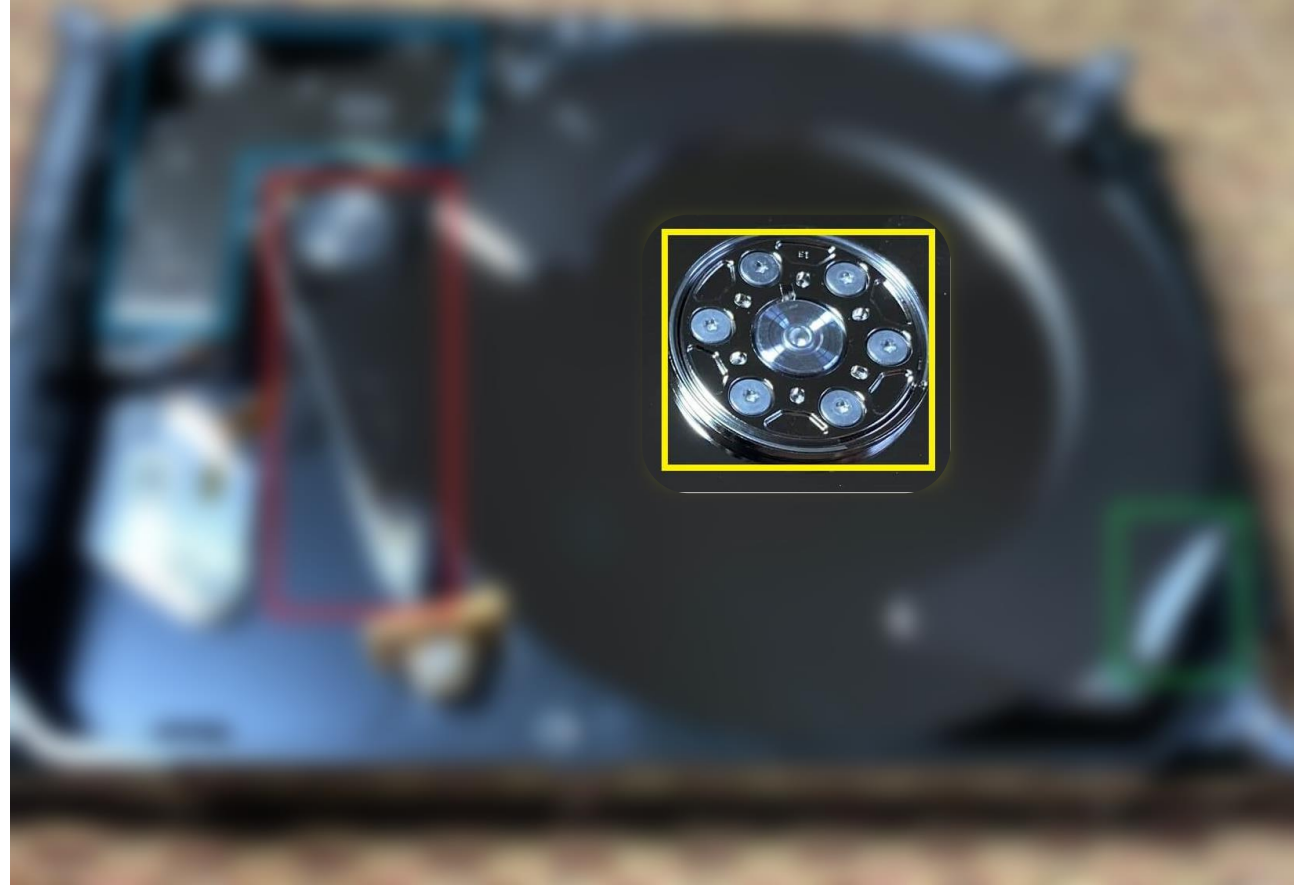
Magneet

- Wordt gebruikt om arm te laten bewegen
- 2 per HDD



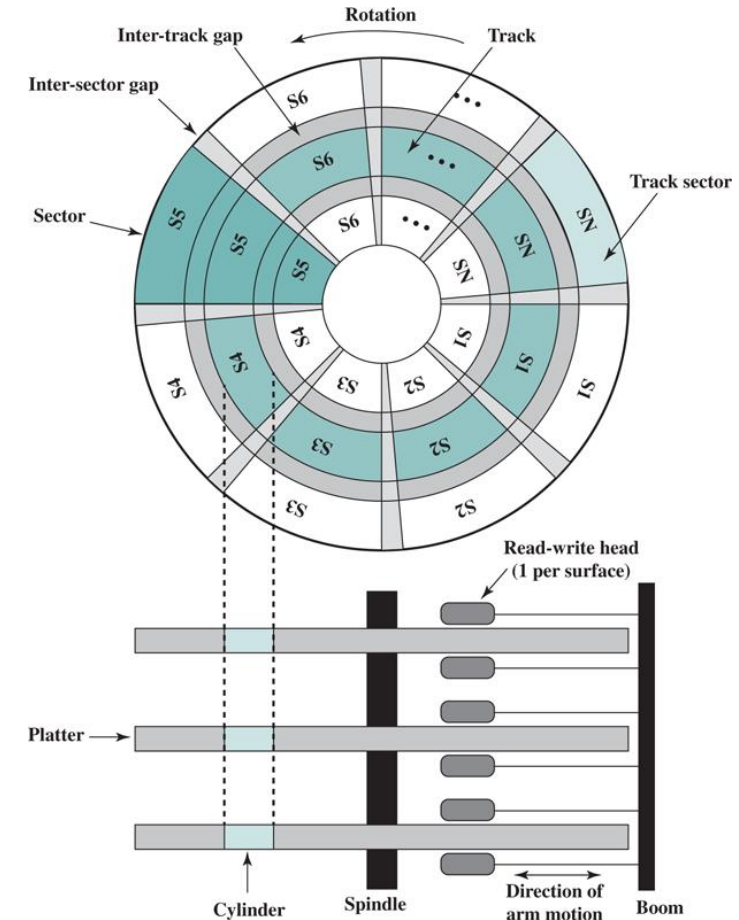
Draaimotor

- Laat schijven draaien
- Courante snelheden:
 - 5400 RPM
 - 7200 RPM



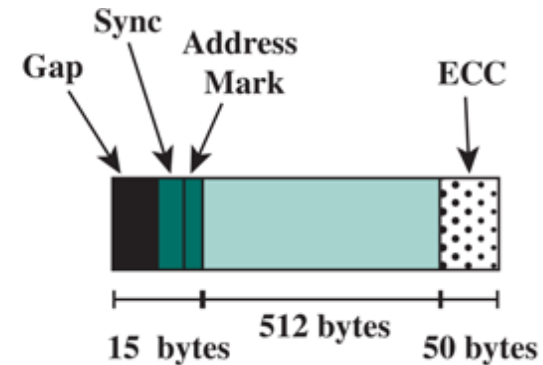
Sporen en sectoren

- HDD bestaat uit
 - Sporen:
 - Rij van bits
 - Lineaire dichtheid (aantal bits/lengteenheid). 50000 bits/cm
 - Sectoren:
 - Deel van schijf dat begrensd wordt door 2 stralen
 - Cilinder:
 - Alle sporen onder elkaar (of onder de koppen van de arm)

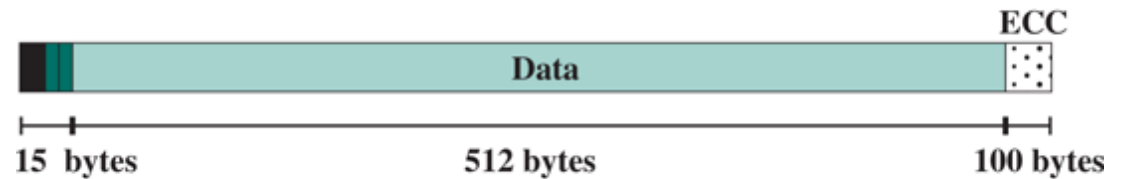


Opbouw sector

- Vaste grootte
 - 512 bytes
 - 4096 bytes (sinds 2010)
- Vaste structuur:
 - **Gap**: scheidt sectoren van elkaar
 - **Sync**: begin van sector, zorgt voor uitlijning
 - **Address Mark**: bevat sectornummer



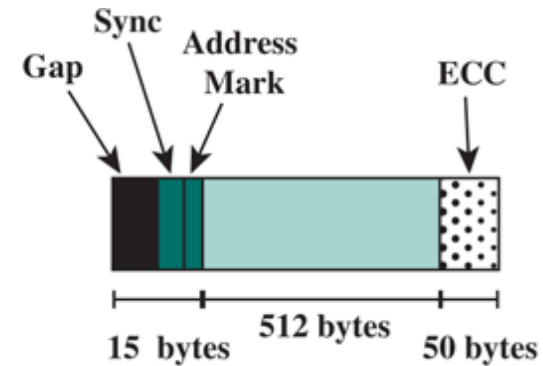
(a) Legacy 512-byte sector



(b) Advanced Format 4k-byte sector

Opbouw sector

- Vaste structuur (part 2):
 - **Data**: eigenlijke data dat moet opgeslaan worden
 - **ECC**: aantal bytes om data uit sector te kunnen recuperen



(a) Legacy 512-byte sector



(b) Advanced Format 4k-byte sector

Sectornummer

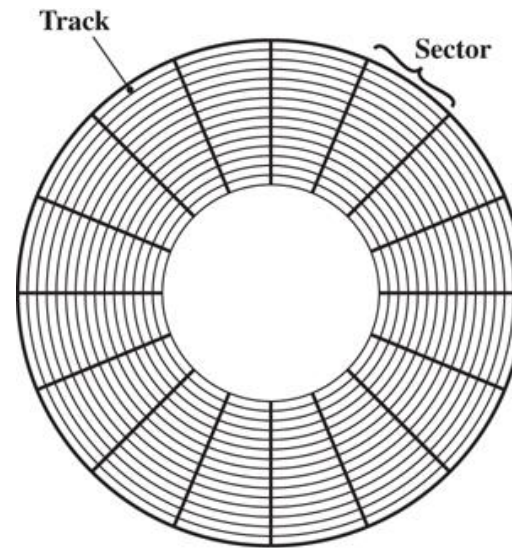
- OS houdt voor elk bestand bij in welke sector(en) dit zich bevindt:
 - Vroeger: CHS (Cilinder, Head, Sector)
 - Nu: LBA (Logic Block Addressing)
 - Chronologische nummering begint bij 0
 - Eerst 22 bit mode voor adressen
 - Maximum capaciteit?
 - $2^{22} * 512 \text{ bytes} = 2 \text{ GB}$
 - Nu 48 bit mode voor adressen
 - $2^{48} * 512 \text{ bytes} = 144 \text{ PB}$

Magneetschijven: capaciteit

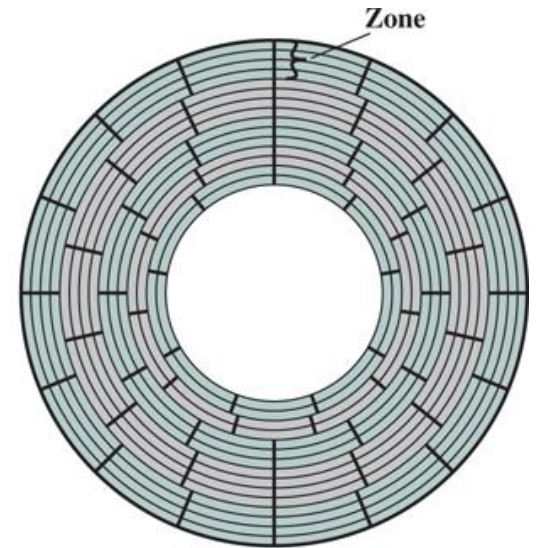
- Capaciteit is het aantal bytes dat beschikbaar is.
- **Netto-capaciteit**: het aantal bytes dat voor data kan gebruikt worden
- **Bruto-capaciteit**: netto-capaciteit, plus hoofdingen, controle bits en gaps

Zone bit recording

- LBA is de norm
- Niet elk spoor moet hetzelfde aantal sectoren hebben
- Buitenste spoor langer:
 - Veel meer bits
- Optimaal gebruik?
 - Meer sectoren naarmate we naar buitenste spoor gaan = **Zone bit recording**



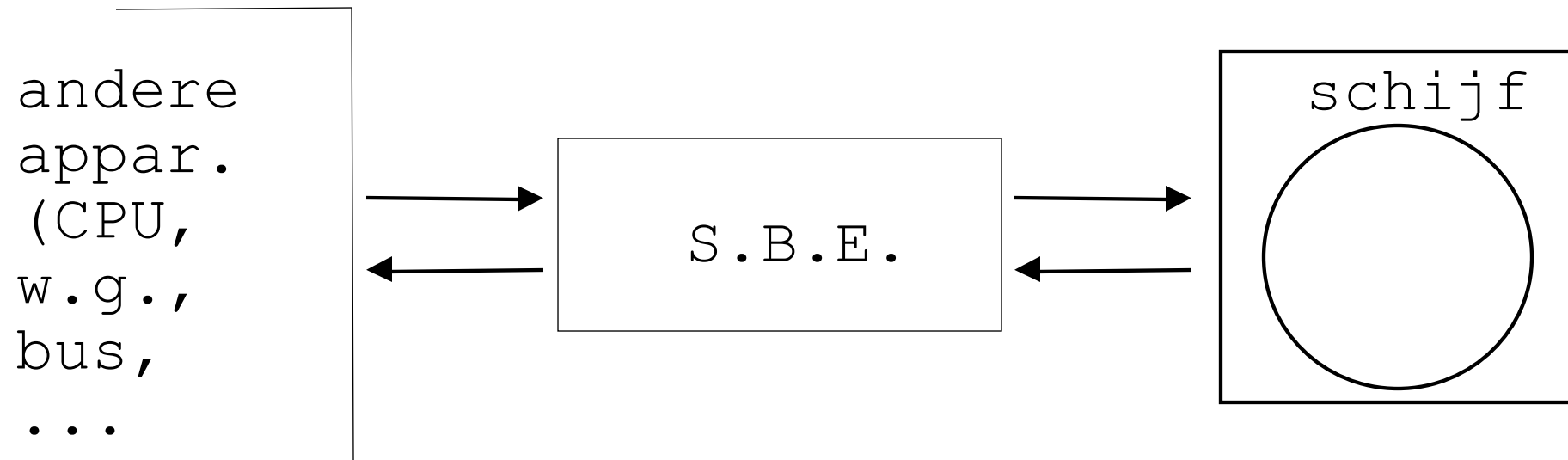
(a) Constant angular velocity



(b) Multiple zone recording

Schijfbesturingseenheid

- Schijf (incl. de aandrijfeenheid) communiceert met de rest van de apparatuur via de S.B.E.



Wat doet de SBE?

- Arm bewegen tot kop op het juiste spoor staat (**seek**)
- Van alle sectoren de hoofding lezen, tot juiste sector passeert (**search**)
- **Schrijven:**
 - Gegevens (bvb. 4 kbyte) aannemen van de CPU en opslaan in buffer (in SBE)
 - **Wegschrijven naar schijf**
 - elektrische informatie (0V/5V) → magnetische informatie (noord/zuid)

Wat doet de SBE?

- Arm bewegen tot kop op het juiste spoor staat (**seek**)
- Van alle sectoren de hoofding lezen, tot juiste sector passeert (**search**)
- **Lezen**
 - Gegevens **lezen van schijf** en opslaan in buffer (in SBE)
 - magnetische informatie (noord/zuid) → elektrische informatie (0V/5V)
 - Gegevens (4 kbyte) naar de CVE sturen

Wat doet de SBE?

- Pariteitsbits berekenen en toevoegen (bij schrijven)
- Pariteitsbits controleren (bij lezen), eventueel opnieuw lezen
- Clusters niet lezen in de volgorde van de vraag, maar met wachttijden zo klein als mogelijk = native command queuing (NCQ)

SBE: soorten

- Verschillende standaarden
 - **ATA** (IDE, EIDE)
(Advanced Technology Attachment)

DISCONTINUED



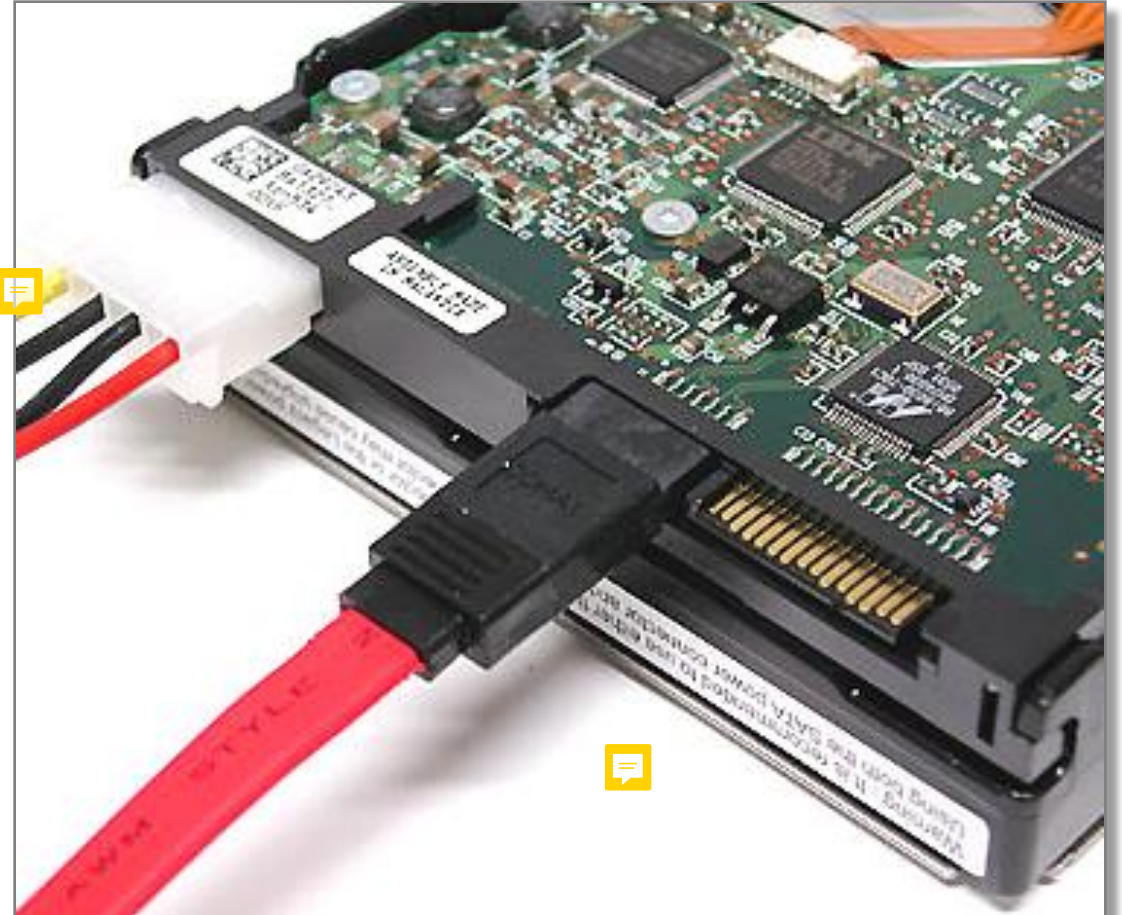
SBE: soorten

- Verschillende standaarden
 - SCSI
 - Duurder dan ATA
 - Vooral bedoeld voor servers
 - Ook serial attached scsi : SAS



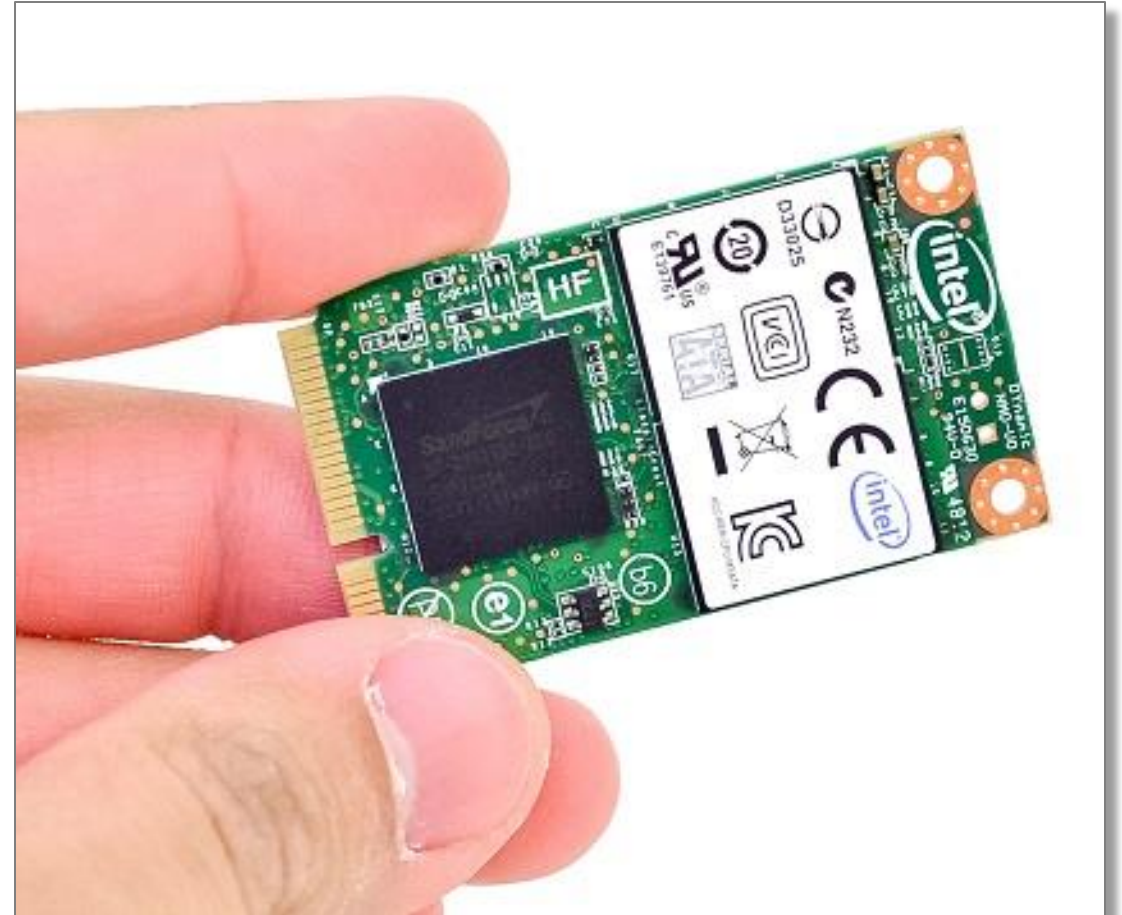
SBE: soorten

- Verschillende standaarden
 - S-ATA
 - Serieel ATA
 - 2 draden voor data, 2 voor besturing
 - Zeer populair voor hedendaagse low-end tot high-end schrijven



SBE: soorten

- Verschillende standaarden
 - M.2
 - Huidige standaard voor zéér snelle schijven
 - Kleine form-factor
 - SATA // NVMe



Cache

- HDD's kunnen traag zijn:
 - Kop moet steeds bewegen naar juiste plaats
 - Wachten tot de juiste sector onder de kop staat
 -

==> Lang wachten
- Oplossing:
 - Snel stukje geheugen in HDD (vergelijkbaar met RAM)
 - Meest gelezen en geschreven data bijhouden in **CACHE**

Disk cache

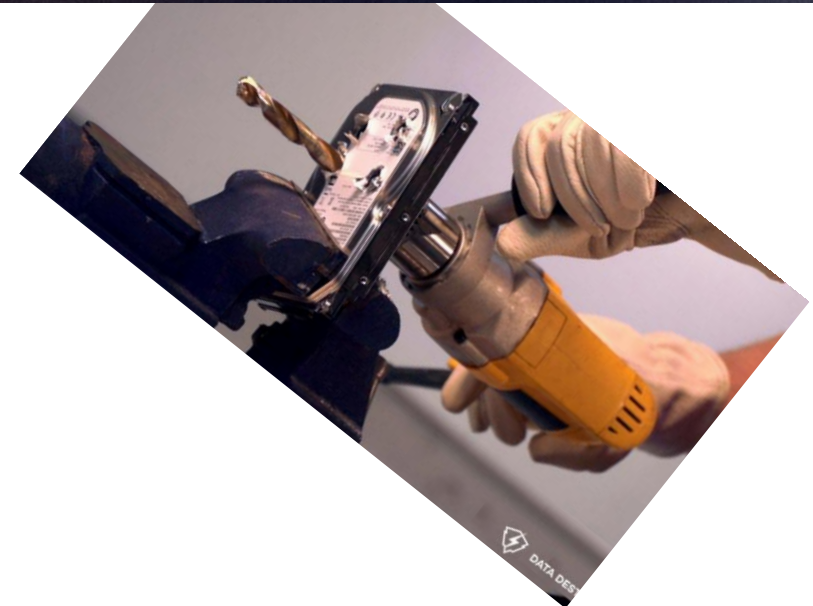
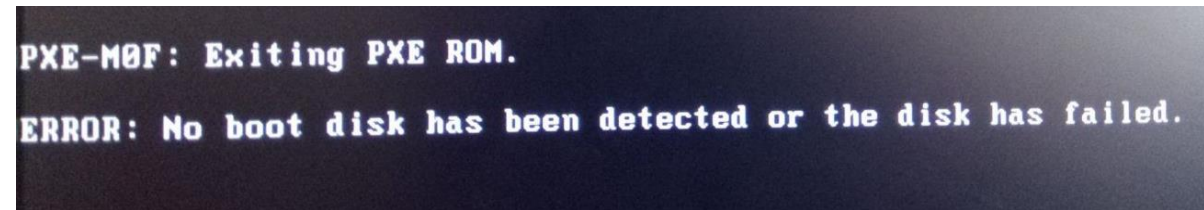
- In de cache-hardware is een algoritme ingebouwd om te beslissen wat in de cache komt
 - De bytes die de CPU gebruikt, staan meestal in mekaars buurt (d.i. **het principe van lokaliteit**). Dus voorspelbaar.
 - Eens je in een bepaald gebied werkt, blijf je daar vaak.
 - Veel gebruikte data in zelfde gebied blijft in cache
 - Eén sector lezen? Kans is groot dat je volgende ook nodig hebt.
 - ==> In cache

Disk cache

- Bestand lezen van HDD?
 - Kans is groot dat deze zal aangepast worden
 - Waarna het moet weggeschreven worden
 - ==> Cache

Raid

- Huidige HDD's /SSD's zijn min or meer betrouwbaar.
- **"Failure is always an option"**
- HDD fabrikanten willen:
 - Tijdwinst voor lezen en schrijven
 - HDD betrouwbaarder maken



Raid

- HDD technologie zit op zijn maximum

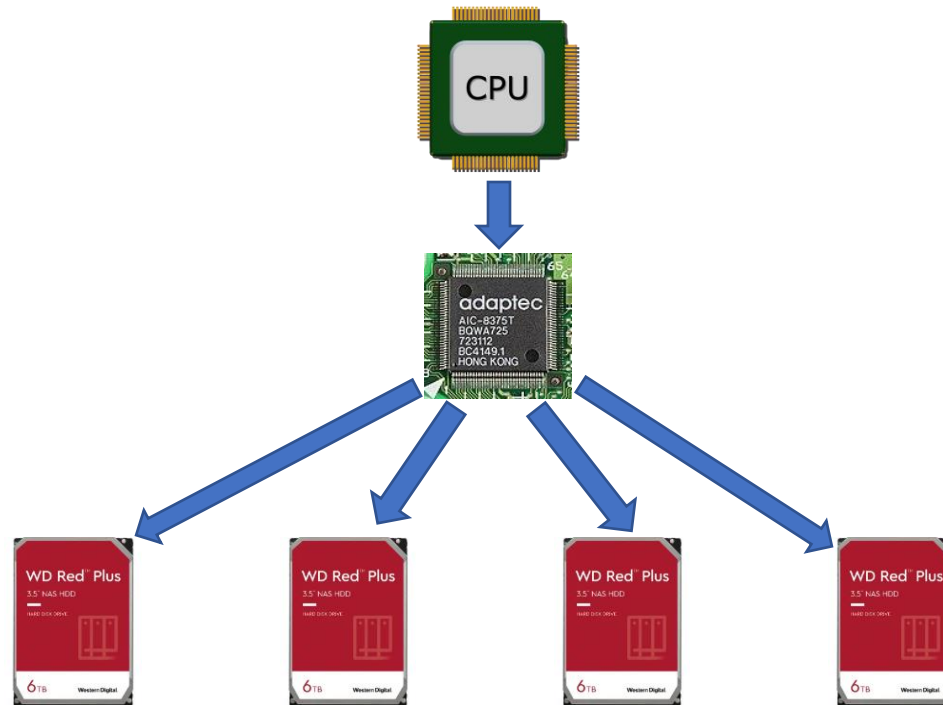
Meerdere HDD's parallel gebruiken voor tijdsinst en betrouwbaarheid.



RAID (Redundant Array of Independent Disk)

Algemeen principe

- DCU heeft meerdere schijven te zijner beschikking, CPU ziet alleen de DCU
- DCU noemen we dan een RAID-controller



Levels

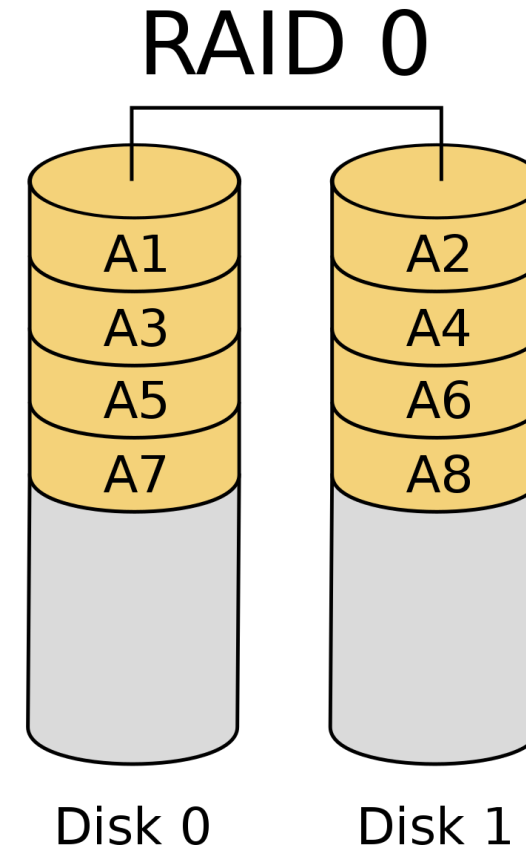
- Raid wordt onderverdeeld volgens verschillende levels:
 - 0 tem 6
- Elke level zorgt voor een betere
 - Betrouwbaarheid
 - of
 - IO tijdswinst
 - of
 - Beide

Raid overzicht

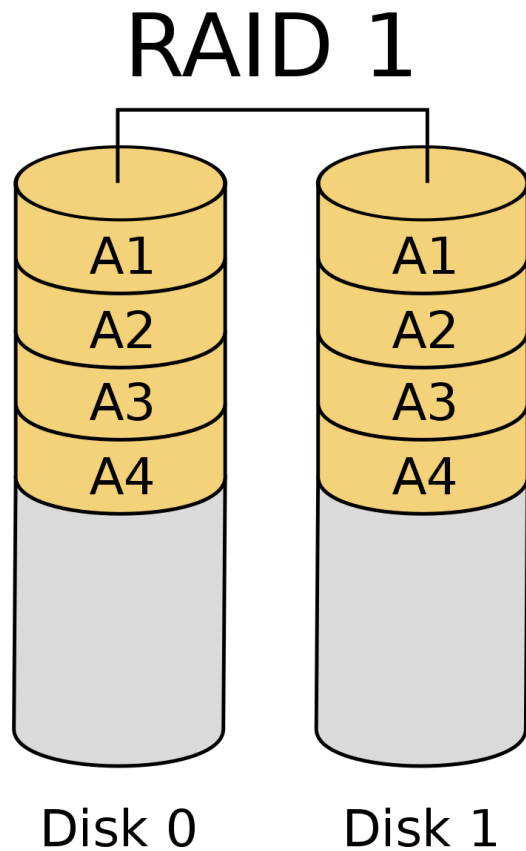
Category	Level	Description	Min. Disks Required	Max. faulty disks
Striping	0	Nonredundant	2	0
Mirroring	1	Mirrored	2	N-1
Parallel access	2	Redundant	3	1
	3	Bit-interleaved parity	3	1
Independent access	4	Block-interleaved parity	3	1
	5	Block-interleaved distributed parity	3	1
	6	Block-interleaved dual distributed parity	4	2

Raid 0

- “Striping”
- Gegevens worden in kleine blokken verdeeld
- Op verschillende schijven geschreven.
- Resultaat:
 - Lezen/schrijven sneller
 - Geen foutcorrectie



Raid 1

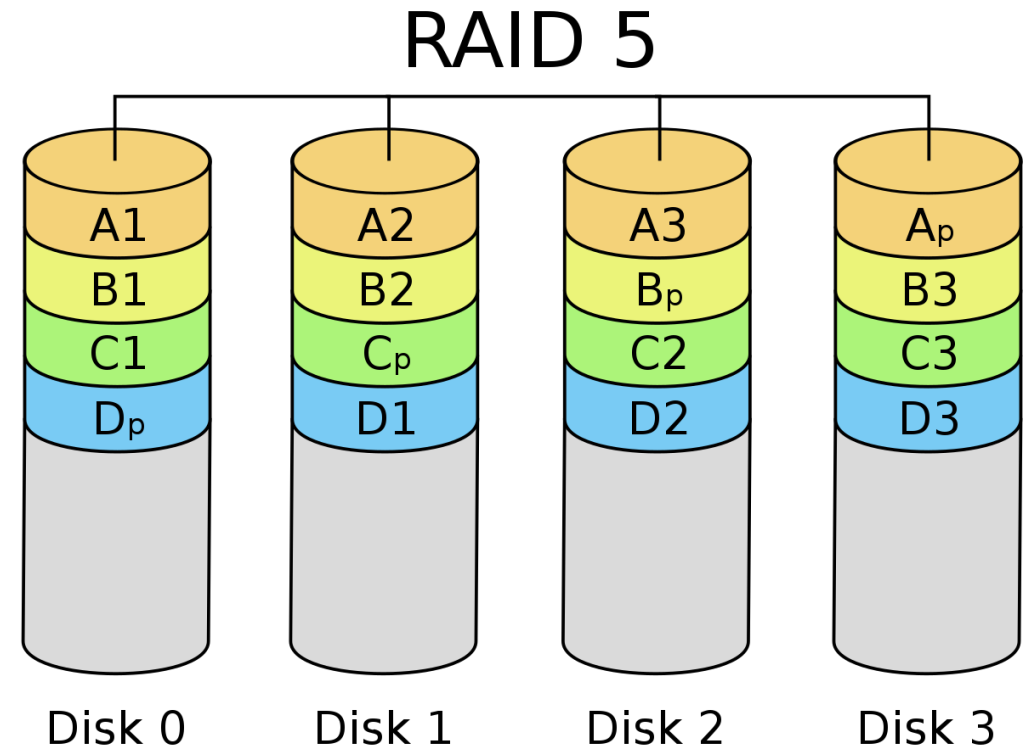


- “Disk mirroring”
- Elke datablok wordt op elke schijf weggeschreven
- 1:1 kopie
- Resultaat:
 - Lezen sneller
 - Schrijven niet sneller
 - Alle schijven buiten 1 mogen stuk gaan.

Raid 5

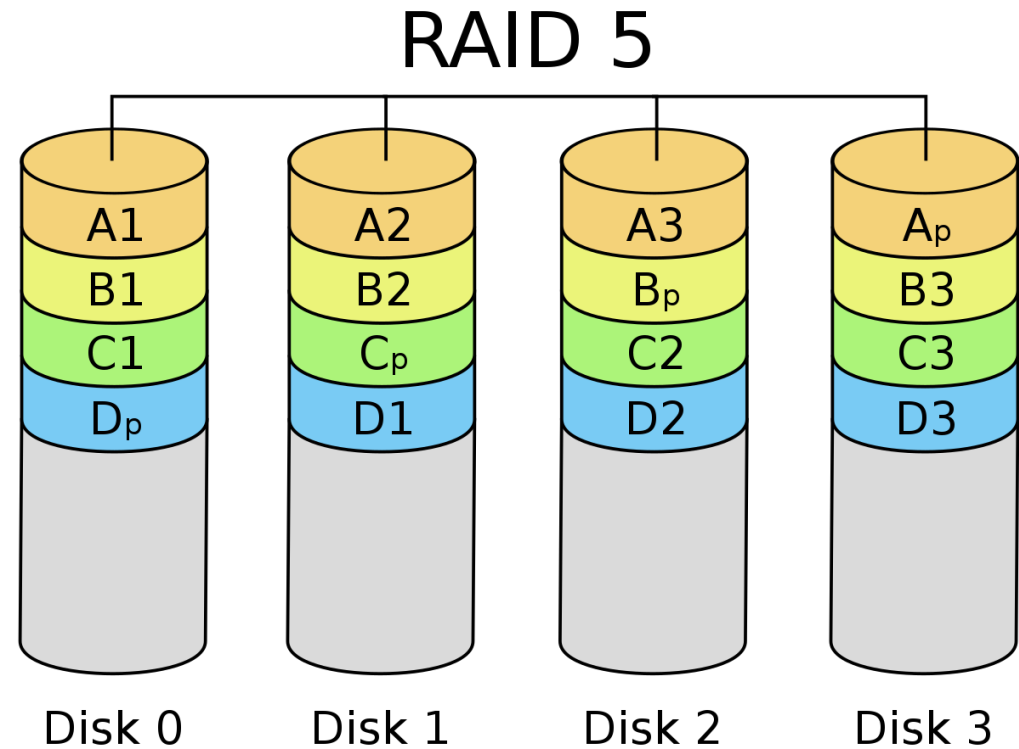
- Data wordt verdeeld in N-1 blokken.
- Van alle blokken wordt pariteitsblok berekent:
 - XOR van alle blokken

```
00101001 (byte 1)
10010101 (byte 2)
00101011 (byte 3)
-----XOR
10010111
```

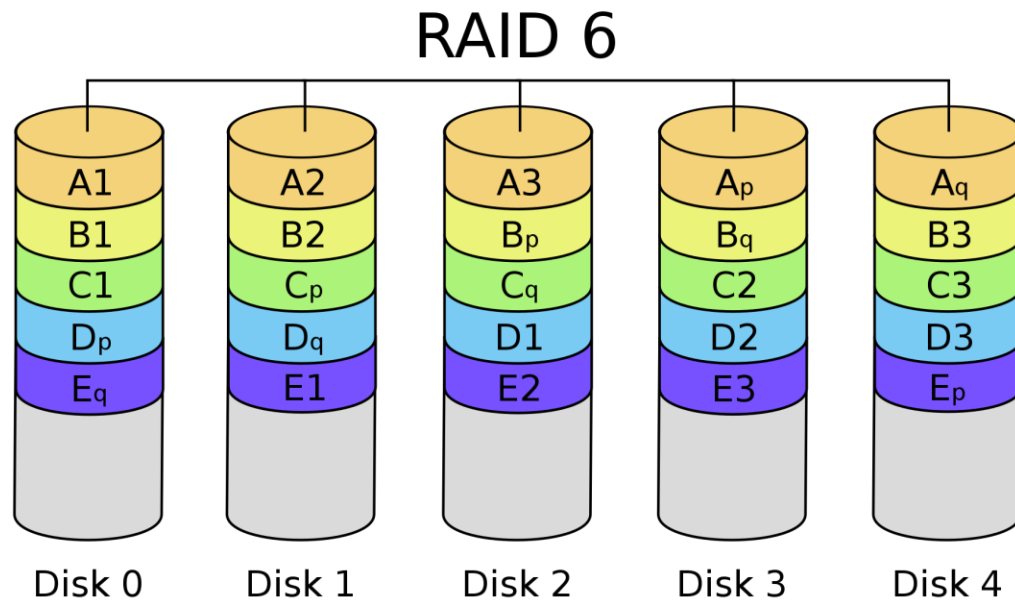


Raid 5

- Pariteitsblok wordt steeds op andere schijf geschreven.
- Resultaat:
 - Lezen en schrijven gaat sneller
 - Bij disk failure: geen data verloren (pariteitsblok)

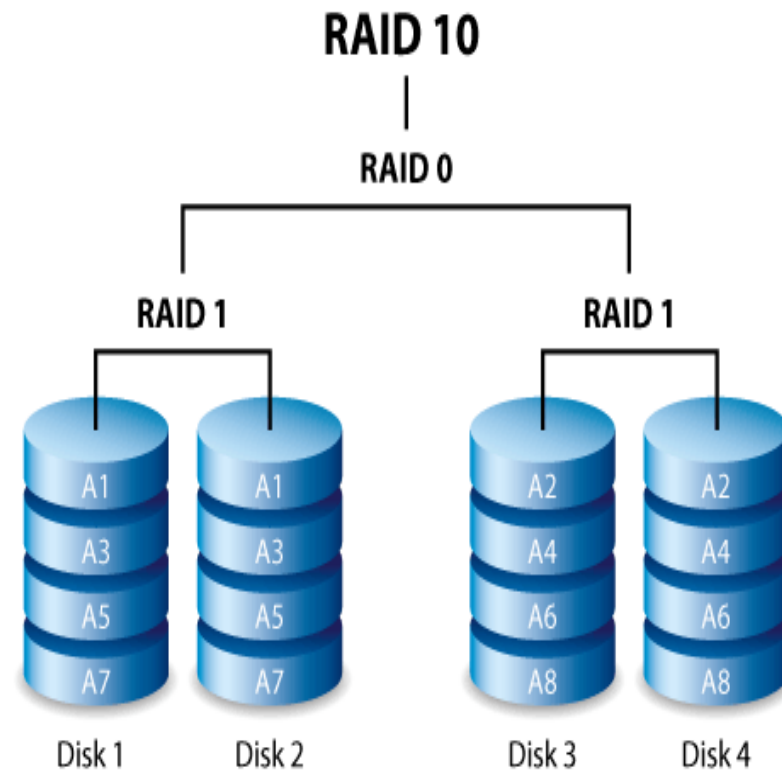
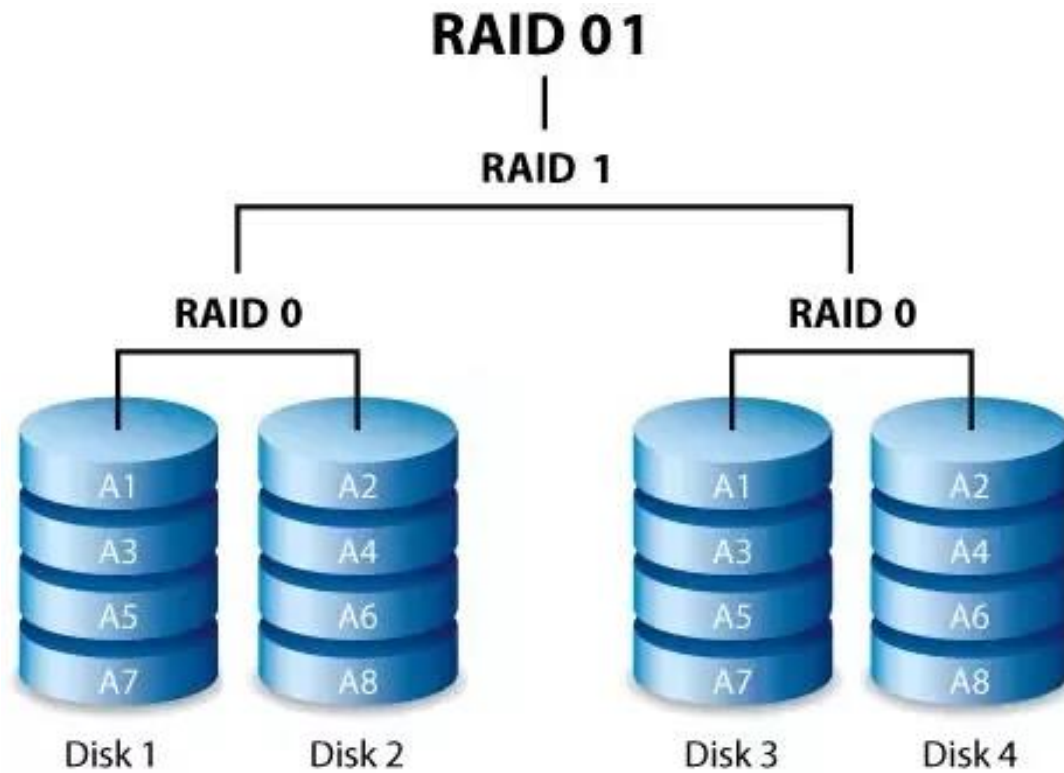


Raid 6



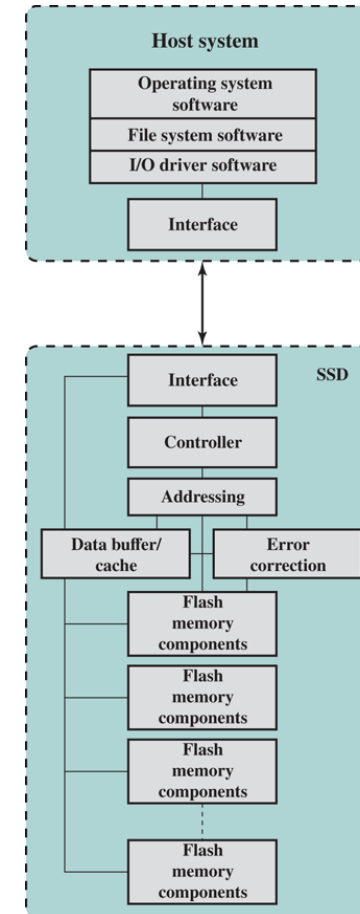
- Per groep blokken 2 pariteitsblokken
- Nadeel: bruikbare ruimte neemt af met twee schijven
- Resultaat
 - Lezen/schrijven sneller
 - Meerdere disk failures toegestaan (max 2 disks)

Combinatie mogelijk



SSD

- HDD's traag (seektime,)
- SSD = “**RAM on steroids**”
- Maakt gebruik van zelfde DCU
- Opgebouwd uit meerdere NAND memory modules die slim samenwerken
- Exacte werking na basisbegrippen Memory



SSD componenten

- **Controller**

- gebruikt specifieke firmware.

- **Addressing**



- Omzetting van adressen OS naar aanspreking NAND flash chips.

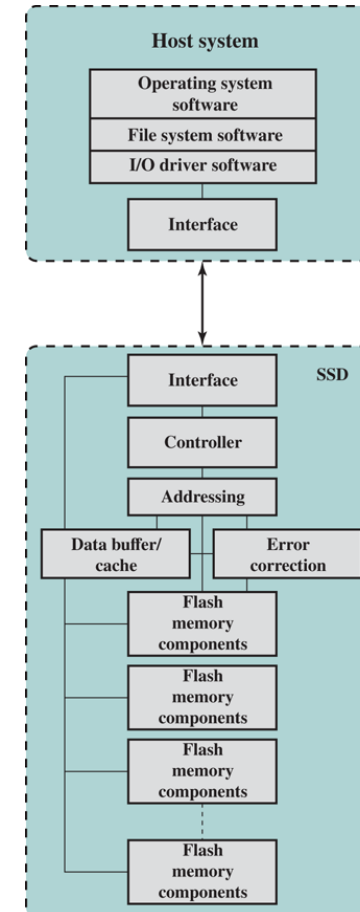
- **Data buffer/cache**

- **Error correction**

- Berekening error detection en correction

- **Flash memory components**

- De individuele NAND flash chips.



SSD: Voordelen

- **Hoge IOPS** (input/output operations per second)
 - Een SSD haalt veel hogere lees en schrijfsnelheden.
 - Samsung 870 EVO 1TB heeft 560 MB/s aan leessnelheid.
 - WD Red Plus 1TB heeft 150 MB/s aan leesnelheid.
- **Duurzaam:**
 - Enkel elektronische deeltjes, geen mechanische
 - Geen invloed van schokken en vibraties

SSD: Voordelen

- **Langere levensduur:**
 - Geen mechanische slijtage.
- **Lage Power Consumption:**
 - SSD verbruikt minder energie
 - Samsung 870 EVO 1TB verbruikt 2W
 - WD Red Plus 1TB verbruikt 3,3W.

SSD: Voordelen

- Lagere toegangstijd en latency
 - 5 keer sneller aanspreekbaar dan HDD
 - Niet wachten op arm die op juiste plek staat

SSD: nadelen

- SSD wordt trager
 - Hoe meer men een SSD gebruikt
 - SSD maakt gebruik van paging voor opslaan van data
 - Zeer slim gebruik, maar maakt SSD trager na verloop van tijd
 - Wordt verder in OPO grondig besproken



SSD: nadelen

- SSD heeft een eindig bestaan:
 - Wordt onbruikbaar na x aantal schrijfoopdrachten
 - Uitgedrukt **Terra Byte Written = TBW**
 - Standaard SSD in laptops heeft TBW van 60 tot 150
 - 190GB schrijven per dag gedurende 1 jaar.
 - Berekening zou je ondertussen zelf moeten kunnen doen



GPT

- Wat als je 2 OS'en op 1 HDD/SSD wil installeren?
- Wat als je maar 1 HDD/SSD hebt en toch meerdere logische schijven wil?
 - ➔ Partities
 - ➔ **GUID Partition Table (GPT)**

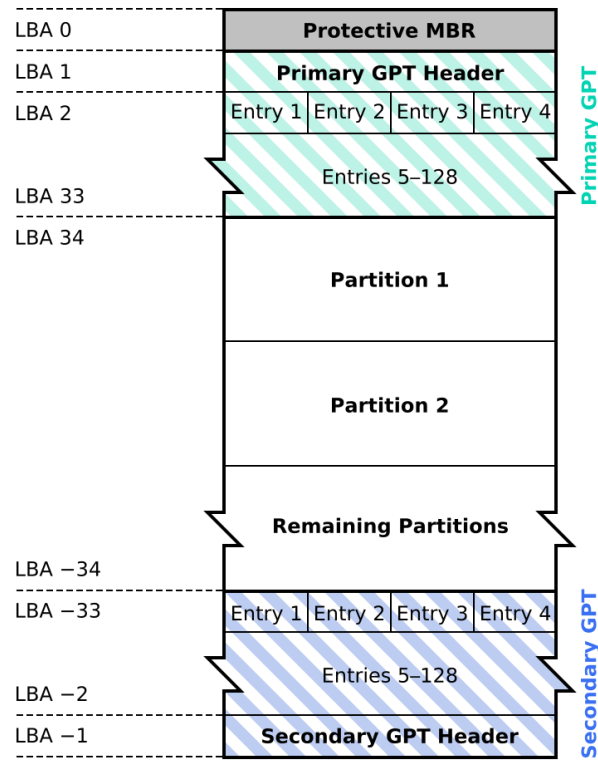
GPT: principe

- GPT = GUID Partition Table
- GUID = UUID = Universal Unique Identifier
 - 128 bit uniek label
 - Elke harde schijf krijgt UUID
 - Elke partitie krijgt UUID
 - Wordt door OS gebruikt om deel of gehele harde schijf aan te spreken
- HDD/SDD verdelen in logische blokken
 - Vrije keuze van grootte
 - Blokken hebben geen toegang tot elkaar
- Enkel mogelijk in UEFI (Legacy BIOS gebruikt MBR voor partities)



Partition Table


GUID Partition Table Scheme



- GPT schema bestaat uit:
 - GPT header
 - GPT entries
- Staat steeds aan begin van HDD/SDD
- Altijd exacte kopie op einde van HDD/SDD ==> Backup voor faulty sectors

GPT Header

Offset	Length	Contents
0	8 bytes	Signature ("EFI PART", 45 46 49 20 50 41 52 54)
8	4 bytes	Revision (For GPT version 1.0 (through at least UEFI version 2.3.1), the value is 00 00 01 00)
12	4 bytes	Header size in little endian (in bytes, usually 5C 00 00 00 meaning 92 bytes)
16	4 bytes	CRC32 of header (0 to header size), with this field zeroes during calculation
20	4 bytes	Reserved; must be zero
24	8 bytes	Current LBA (location of this header copy)
32	8 bytes	Backup LBA (location of the other header copy)
40	8 bytes	First usable LBA for partitions (primary partition table last LBA + 1)
48	8 bytes	Last usable LBA (secondary partition table first LBA – 1)
56	16 bytes	Disk GUID (also referred to as UUID on <u>UNIXes</u>)
72	8 bytes	Partition entries starting LBA (always 2 in primary copy)
80	4 bytes	Number of partition entries
84	4 bytes	Size of partition entry (usually 128)
88	4 bytes	CRC32 of partition array
92	*	Reserved; must be zeroes for the rest of the block (420 bytes for a 512-byte LBA)
LBA size	Total	

- Steeds op LBA 0 
- Bevat alle GPT belangrijke info:
 - Signature (Steeds "EFI PART")
 - Current LBA (Locatie van GPT tabel)
 - Backup LBA (locatie van backup GPT info)
 - Aantal partities in de tabel
 - Grootte van item in partitietabel

GPT entry

GUID partition entry format

Offset	Length	Contents
0 (0x00)	16 bytes	Partition type GUID (mixed endian ^[7])
16 (0x10)	16 bytes	Unique partition GUID (mixed endian)
32 (0x20)	8 bytes	First LBA (little endian)
40 (0x28)	8 bytes	Last LBA (inclusive, usually odd)
48 (0x30)	8 bytes	Attribute flags (e.g. bit 60 denotes read-only)
56 (0x38)	72 bytes	Partition name (36 UTF-16LE code units)

- Opeenvolging van details over partities:
 - Partition type GUID
 - Vooraf bepaalde GUID
 - Type = boot, swap, root, home, ...
 - UUID van partitie
 - Start en laatste LBA van partitie
 - Attributen
 - Zelf gekozen naam van partitie

Bestanden

Fysische ordening

- Sector
 - Kleinste adresseerbare eenheid
 - 512 bytes
 - Heeft uniek adres = LBA

Logische ordening

- Filestysteem:
 - Bestand = basiselement
 - Slim gebruik van fysische sectors
 - OS gaat hier mee aan de slag
 - Vooraf gedefiniëerde gegevensstructuur

Blokken

- Kleinste bruikbare eenheid = sector (512 bytes)
- Filestysteem groepeert sectoren in grotere eenheid: blokken
 - Vb: blocksize van 4096 bytes = $8 * 512$ sector
 - Voor OS is blocksize kleinste adresseerbare eenheid.

Filesystemen

- Mens gebruikt bestanden \Leftrightarrow computer gebruikt bits en logica
 - Elk bestand neemt X aantal blokken in
 - Bestandsomschrijving van elk bestand:
 - Naam
 - Aantal blokken, verwijzing naar correcte aantal blokken, ... (elk Filestysteem anders)
 - Timestamps
 - Size
 - Rechten
 - ...
- Meest gebruikte filesystemen:
 - FAT
 - NTFS
 - EXT

FAT

- File Allocation Table
 - File map: informatie van elk blok
 - Verschillende mogelijkheden
 - Onderdeel van bestand?
 - Adres van volgende blok
 - Laatste blok = EOF code in de fat
 - FAT12, FAT16, FAT32
 - Aantal bits die gebruikt worden in FAT entry
- USB sticks worden nog steeds standaard geformatteerd met FAT32

FAT

- Enkel 'linked' lijst niet voldoende:
 - Weten enkel of blok gebruikt is of niet
 - Correlatie nodig met bestanden/directories en gebruikte blokken

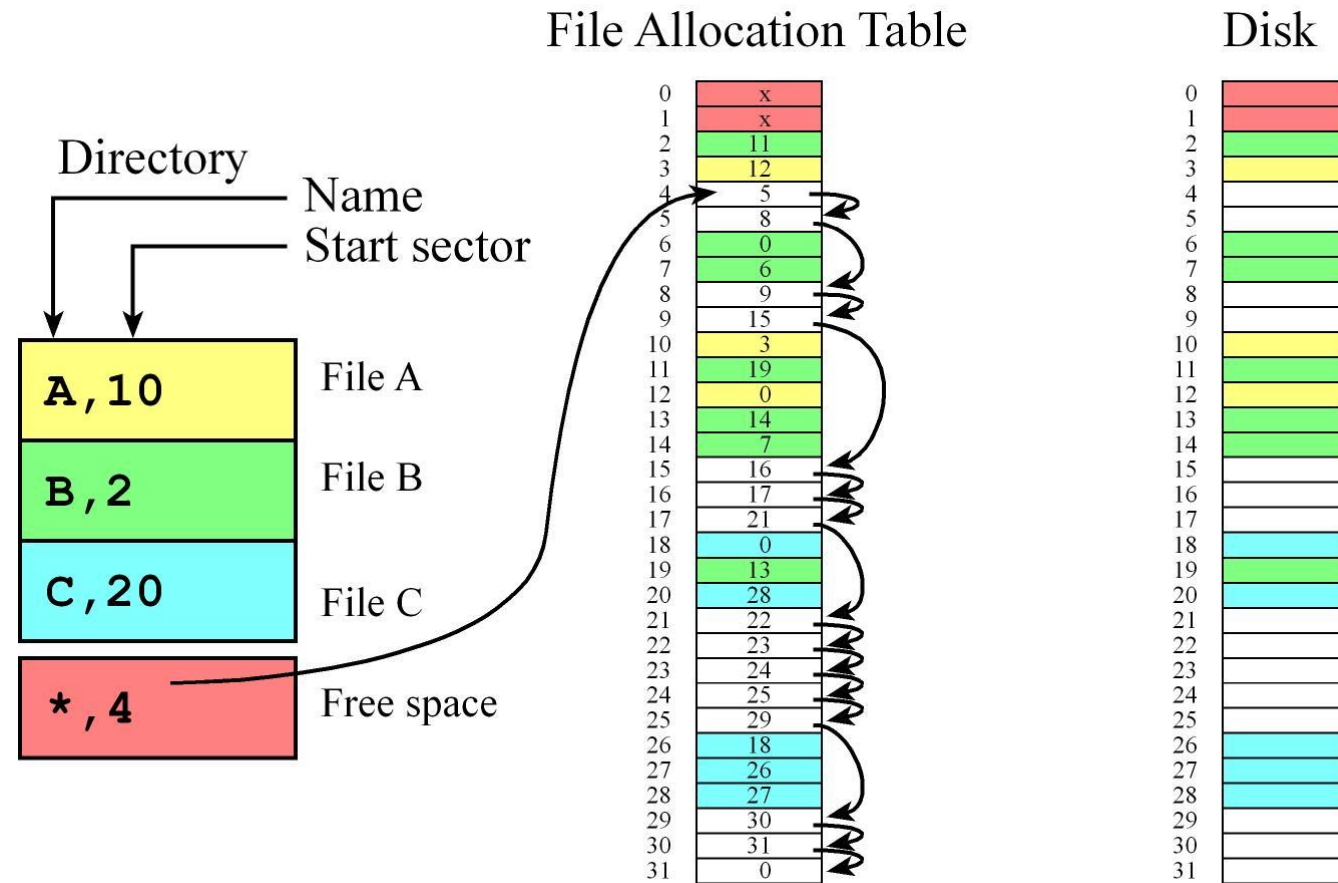
==> Directory Table

Directory Table

- Meerdere tables
- Bevat informatie over:
 - Naam bestand of directory
 - Extensie
 - Eerste blok van bestand of eerste blok van volgende directory table

Veld	Positie	Lengte
Naam	0	8
Extensie	8	3
Attributen	11	1
Tijdstip Creatie	12	1
Hoogste bytes startadres	20	2
Startadres (FAT32: laagste bytes)	26	2
Grootte	28	4

FAT



NTFS

- New Technologie File System
- Ontwikkeld door Microsoft
- Standaard filesysteem vanaf WinXP
- Maakt gebruik van Master File Table

MFT

- Per bestand, 1 record in MFT
 - Initieel 12,5% van HDD
 - Kan groter worden indien meer bestanden
 - Overige 87,5 % van HDD voor data
 - Data ruimte vol?
 - MFT ruimte nog niet vol?
- ==> Data in MFT blok bewaren (niet aan te raden)



MFT

Header	Standard Information	Attributes	Filename	Data	...	Security Descriptor
--------	----------------------	------------	----------	------	-----	---------------------

- Steeds vaste structuur:
 - 1024 bytes
 - **Header**
 - **Standard Info:** Bevat onder andere grootte, timestamps, ...
 - **Attributen:** Bitmap voor attribute-file in begin van MFT tabel en extra attributen.



MFT

Header	Standard Information	Attributes	Filename	Data	...	Security Descriptor
--------	----------------------	------------	----------	------	-----	---------------------

- Steeds vaste structuur:

- **Filename**

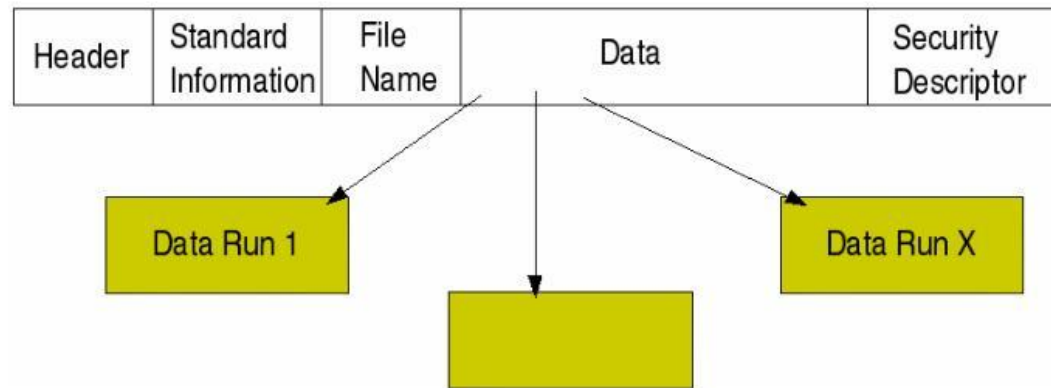
- **Data:**

- Bestand < 900 bytes, in MFT entry zelf
 - Bestand > 900 bytes, verwijzing naar datablokken van bestand

- **Security descriptor:** Alle rechten op bestand

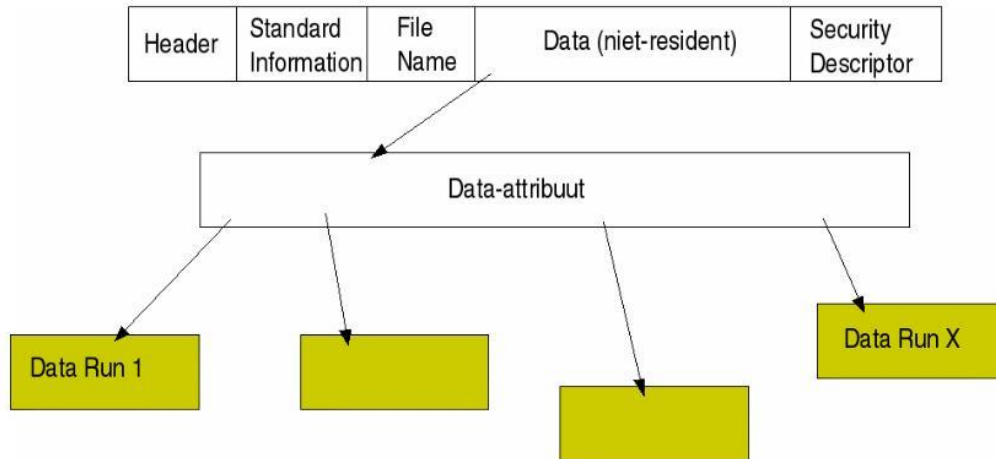
MFT: data

- Enkelvoudige verwijzing naar datablokken voor bestand



==> 'Data runs' = term van Microsoft

MFT: Data



- Te weinig plaats voor dataruns?
 - Verwijzing naar 'speciaal' datablok
 - Alleen maar data-attributen
 - Of verwijzing naar nog data-attributen blokken

==> Max filesize = 'Partitie grootte'

OS bepaalt zelf limiet

EXT

- Extended File System
- ext2, ext3, **ext4**
- Standaard filesystem voor linux
- Maakt gebruik van 4 standaardblokken:
 - Bootblok
 - Superblok
 - Inodeblokken
 - Datablokken

Ext

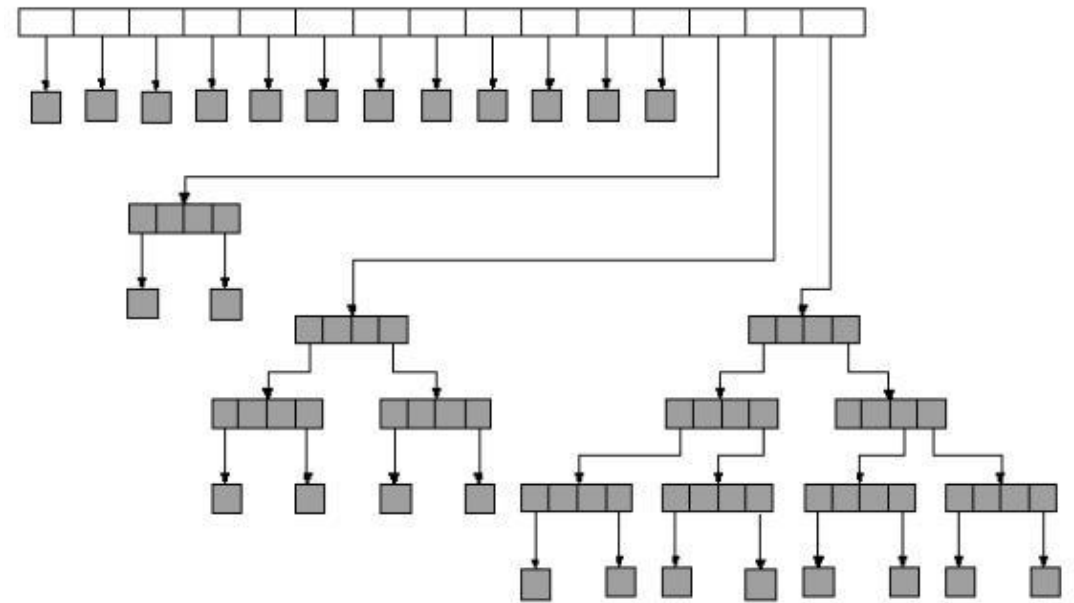
- Bootblok:
 - Eerste blok van schijf:
 - Geen functie in filesystem
 - Enkel code die OS laadt en opstart
- Superblok
 - Globale gegevens
 - Algemene instellingen
- Inodeblokken
- Datablokken

Inodeblok

- Per bestand 1 Inode blok:
 - Alle metadata van bestand
 - Verwijzing naar datablokken die gebruikt worden voor bestand
- Bestandsnaam staat **NIET** in de inode
- Vast aantal inodes
 - Wordt bepaald bij aanmaak bestandsysteem
 - ==> max aantal bestanden

Inode

- Ruimte voor 15 adressen:
 - 12 directe adressen
 - 1 Indirect adres
 - 1 dubbel indirect adres
 - 1 driedubbel indirect adres
- Max file size = 16 TB (4K blocks)



Ext

- Filename?
 - Staat in directory entry
- Directory:
 - Is eigenlijk een file
 - Lijst (directory entries) met filename
 - Correcte verwijzing per file naar inode

Ext

- Welke inodeblok nemen bij nieuw bestand?
 - Superblok bevat lijst 100 lege inode blokken
 - Superblok bevat lijst lege datablokken
- Bestand verwijderen:
 - Inode en datablok adres toevoegen aan superblok lijst.
- Lijsten leeg in Superblok:
 - Kernel scant systeem voor lege inode blokken en datablokken
 - Vult superblok lijst aan