

# BIBELEN



## Innhold

<b>1</b>	<b>PC Komponenter</b>	<b>1</b>
1.1	Servere . . . . .	1
1.2	Rack-skap . . . . .	1
1.2.1	Blade-servere . . . . .	1
1.3	Ulike PC komponenter . . . . .	1
1.3.1	Hovudkort . . . . .	1
1.3.2	Chipset . . . . .	1
1.3.3	RAM . . . . .	1
1.3.4	CPU . . . . .	2
1.3.5	Harddiska . . . . .	2
1.3.6	Nettverkskort . . . . .	2
1.3.7	Lagringkontroller . . . . .	3
1.3.8	GPU . . . . .	3
1.4	Latency og Bandwidth . . . . .	3
1.5	Størrelsa . . . . .	3
1.6	Tilkobling av kort . . . . .	4
<b>2</b>	<b>Linux</b>	<b>4</b>
2.1	Oppbygging . . . . .	4
2.2	Katalogtre . . . . .	5
2.2.1	Filsystem . . . . .	5
2.2.2	Brukere/grupper . . . . .	5
2.2.3	Rettigheter . . . . .	6
2.2.4	Kommands . . . . .	6
2.3	SSH . . . . .	7
2.4	Enheter . . . . .	8
2.4.1	Loop-enheter . . . . .	8
2.5	Raid . . . . .	8
2.5.1	Software raid . . . . .	9
2.6	Scripting . . . . .	10
2.6.1	Variabler . . . . .	10
<b>3</b>	<b>Nettverk</b>	<b>11</b>
3.0.1	Standar . . . . .	11

## Figurer

1	Kjernen . . . . .	5
2	Raid typer . . . . .	8

## Listinger

# 1 PC Komponenter

Kva er det som egentlig går under PC komponenter?

1. Server
2. Klient
3. Switch
4. Ruter
5. Kabler
6. Interne komponenter

## 1.1 Servere

## 1.2 Rack-skap

Eit rack skap refererer som regel til eit skap som er produsert i henhold til "19" rack" standar. Der 19" refererer til bredden på skapet. Måle standaren i skap blir ikkje målt i dybde eller høyde, men i "Units". Der  $1U = 1,75''$ , Serverer pleier å ha ein størrelse på 1-4U.

### 1.2.1 Blade-servere

Disse blei først lansert i 2003/2004 av IBM. Dei var tenkt som åpne standarder og alt i serveren er Hot Swap på. Hot Swap vil sei at utstyret kan byttast ut mens serveren kjører.

fordelen med desse serverane er at dei reduserer kostnaden per server og da er enklare å bytta ut heile serveren om da trengs.

Ulempen er at dei har redusert plass i kabinettet og er mindre fleksible i formfaktor

## 1.3 Ulike PC komponenter

På ein PC har enn eit par ulike komponenter Hovudkort, CPU, RAM, Harddisk, Nettverkskort og Skjermkort.

### 1.3.1 Hovudkort

Formålet med Hovudkort er at da skal kobla alle komponentane saman, da skal styra strøm til kvar av komponentane og da skal håndtera oppstarten av PCen. Meste parten av hovudkort idag har ein del utstyr som er integrert i kortet, som f.eks. nettverkskort, skjerkort og lydkort

### 1.3.2 Chipset

Er ein sentral brikke på hovudkortet. Den inneheld southbridge og funksjona frå northbridge som ikkje er integrerte i prosessoren. Southbridge handtere som oftast I/O einheta. hos Interl blir denna kalla "I/O Controller Hub" medan hos AMD kallar dei den for "Fusion Controller Hub" Northbridge er som oftast den som tar seg av litt raskare komponentar, som CPUen, RAM, og PCI-E. Den blir ofta kalla "memory controll hub" men da er ingen industriell standar så intel og AMD kan har 2 forskjellige meininga om kva southbridge og northbridge skal ta seg av. Men ein ting er begge enige om og da er at Southbridge ikkje har direkte kontakt med CPUen, men er knytta til Northbridge som igjen er knytta til CPUen.

### 1.3.3 RAM

RAM er ein midlertidig arbeidsdata, ingenting blir lagra her. Den er Volatile nake som betyr at alt blir sletta når den mister strømmen. Den er rask og dyr. Da er ulike standardar: DDR/DDR2/DDR3/DDR4 Da er hovudkortet og prosessoren som bestem kva type som kan bli brukt.

Hastigheten i RAM blir målt i GBps og latencyen blir målt i nano sekunder. Størrelsen blir målt i GB. i dag har ein som oftast ha 4 modular og 32GB til saman.

RAM som er beregna på servera har støtte for ECC(Error-Correcting Code). Den retter korrupsjon av data(bitfeil). Denna korrupsjonen skjer pga elektriske eller magnetiske forstyrrelsa. Dyrare en RAM uten ECC pga ekstra elektronikk på brikka(har 9 istaden for 8).

64bit bus betyr 64 linjer i parallel: 8 Bytes overføres om gangen. Mange hovedkort / CPUa støtter dual channel: bruker to spor samtidig for 16 Byte/ 128 bit overføring om gangen

Dagens RAM er begrensa av avstanden mellom CPU og RAM. Lange linjer forhindrer høgare frekvensar. Dette fordi signalet til lange linjer forsyrrer lettare. Da betyr at da kreves også høgare spenning. RAM har også begrensa plass på hovedkortet.

I framtid så blir RAM plassert direkte på CPUen, eller rett ved siden av i samme "pakke". Mulighet for fleire linjer med raskare frekvens(dobbel bonus). Mulighet for lavare latency. Frir opp plass på hovedkortet - mindre telefonar / tablets / laptopar. Desse er allereie i produksjon for mobile CPUer - ARM (Texas Instruments, OMAP) Er også i produksjon for skjermkort - AMD HBM(2)(256GBps, 10x GDDR5) - Radeon RX VEGA, Radeon VII. Desse er dyrare å produsere i dag nåke som har ført på satsing av GDDR5X og GDDR6 istanden. DDR5 kommer sannsynligvis ein gang.

HBM2 ser i hovudsak ut til å bli brukt i skjermkort for Workstation (Business-markedet). Dette vil truleg også gjelda for HBM3.

Skjermkort til "gamers" blir vanlegvis utstyrt med GDDR5X (Nvidia GeForce GTX 1080, GTX 1080Ti, Titan X) eller GDDR6 (Nvidia GeForce RTX 2080 Ti, 2080, 2070, 2060) minne framfor HBM.

### 1.3.4 CPU

Utfører aritmetiske operasjonar. CPUen tar dei fleste avgjørelsane i maskina og program eksekverer / utføres hovudsakleg på CPUen. Den kan innehloda intern RAM aklla cache(mykje raskare enn ekstern RAM, men liten) delt inn i "lag" og heter derfor L1,L2,...,Ln cache. CPU blir målt i antall kjerner og frekvens, type "4 kjerner på 3.0+ GHz". Da er frekvensen som bestemmer antall operasjonar per sekund, og kjernen (og tåder) nestemmer antall operasjonar på ein gang. Foreløpig er da kun inkrementelle forbedringar med klare hindringa(fysisk begrensing). Da jobbes med 3D stacking av chips - nåke som Erlend syns er meget spennande.

### 1.3.5 Harddiska

Er permanente lagrings plasser, f.eks. OS, dokument, spill, osv. Ein har forskjellige harddisker som, Roterande disk - høy latency, mykje lagringsplass og billig. SSD - relativ rask, lav latency, dyr lagringsplass, nåke begrensa lagring. Idag blir desse ofta kobla opp med SATA, mSATA, PCIe eller M.2 hastigheten på harddiska blir målt i MBps / GBps. latencyen blir målt i millisekund(HDD) og mikrosekund(SSD)

Svakheten med HDD er at da er mange mekaniske deler som tar slitasje skadder over tid. Skriveren må roterast rundt viss dataen du treng ligg på andre sida. Så skal du lesa data frå mange ulike stader samtidig går hastigheten drastisk ned.

Svakheten med SSD er at da er ein relativ ny teknologi og er derfor mindre utprøvd enn HDD. Kontrollen (ein liten CPU på SSDen) kan ha feil i firmwaren. alle produsentane har hatt problem siden teknologien kom

- Intel: Feil i firmware gjorde at SSDer kunne bli "bricked" (død)
- Samsung: Feil i firmware gjorde at hastigheten gikk drastisk ned på gamle data
- OCZ: Feil i alt gjorde at feilraten var / er meget høy

I framtida kommer 3D NAND flash, PCM/PRAM og 3D XPoint å bli meir brukt. 3D NAND og 3D XPoint er til salgs i dag, men trengs meir forskingn før da kan bli meir brukt.

NAND flash har ein del svakheta. Teoretisk begrensing på antall skriveoperasjonar. Dette er lite problematisk for arbeidsstasjoner, men mer relevant for servera. Dårligare enn RAM på å lesa små mengde data. MLC: lagrer fleire bits per celle, billigere, mindre robust. Mindre prosessnoder er mindre robuste

### 1.3.6 Nettverkskort

Tilfører fleire nettverksportar til maskina Kan være raskare enn den innebygde (10Gbit) Kan håndtere visse algoritmer for mindre CPU-bruk Hastigha målast i MBit/Gbit, typisk 100Mbit, 1Gbit, 10Gbit,

### 1.3.7 Lagringkontroller

Tilføre fleire porta for harddiska / SSDer (SATA / SAS / M.2) Kan ha funksjona for RAID (meir om da seinare)

### 1.3.8 GPU

Tilfører skjermutganga til maskina Ikkje alle hovudkort/CPU'er har innebygd skjermkort Finnes i alle prisklassar Hastigha målast på samme måte som CPU Frekvens - Antall operasjonar per sekund Antall kjerner (ikkje sammenlignbart på tvers av arkitekturer) Veldig gode på parallele utregninga Brukast nå i servera for databehandling (GPGPU)

## 1.4 Latency og Bandwidth

Latency eller forsinkelse som da hetter på norsk, måler kor lang tid svar på ein kommando tar og er som oftast målt i millisekund. Latency måler KUN svartid og ikkje kor mykje data som kan sendast omgangen. Lavare er betre.

Bandwidth eller Båndbredde, måler kor mykje data som kan flyttast per sekunde og blir målt i Mega Byte per sekund(MBps). Båndbredden seier ingenting om kor lang tid da tar å få første databit. Høgare er betre.

## 1.5 Størrelsa

Byte og bit - da er 8 bit i ein byte. stor B betyr byte MBps / MByte. liten b betyr bit Mbps / Mbit.

Operativ system bruker 2 forskjellige standara. nokre bruykter SI systemet(base 1000) og nokre bruker IEC(base 1024)

SI:

1. Kilo -  $1000$ : Kbit / KB
2. Mega -  $1000^2 = 1\,000\,000$ : Mbit / MB
3. Giga -  $1000^3 = 1\,000\,000\,000$  Gbit / GB
4. Tera -  $1000^4 = 1\,000\,000\,000\,000$  Tbit / TB

IEC:

1. Kibi -  $1024$  Kibibite / Kibit, Kibibyte / KiB
2. Mebi -  $1024^2 = 1\,048\,576$  Mebibite / Mebit, Mebibyte / MeB
3. Gibi -  $1024^3 = 1\,073\,741\,824$  Gibibite / Gibit, Gibibyte / GiB
4. Tebi -  $1024^4 = 1\,099\,511\,628 \cdot 10^{12}$  Tebibite / Tebit, Tebibyte / TeB

OSX og Linux viser typisk lagringsstørrelser i metric / SI systemet <- Linus Torvalds, Europeisk, metrisk Microsoft Windows bruker IEC systemet for å oppgi lagringsplass. <- Bill Gates. Amerikansk, imperisk

Eksempel på størrelsa og hastighet:

- Bredbånd - 5 til 1000 Mbit/s
- 1 MP3 fil - 3-5 MByte
- 3.5" Floppy - 1440 KByte, 240 Kbit/s
- CD - 700 MByte, 1200 Kbit/s
- DVD - 4.7 / 9.4 GByte, 10.5 Mbit/s
- Blu-ray - 25 / 50 GByte, 36 Mbit/s
- 4k Blu-ray - 66 / 100 GByte, 108Mbit/s
- Netflix - 5.8Mbit/s (1080p), 18Mbit/s (4k)

Nettverk i dag er mykje raskare enn når harddiska vart introduserte. HD-video kreve egentlig ikkje så mykje båndbredde og mange av forbrukarane har allereie raskt nok nett til å streama Blu-ray.

## 1.6 Tilkobling av kort

PCI Express er videreutvikling av PCI, men er ikkje bakoverkompatibel. Da baserer seg på seriell overføring i full duplex. Full-duplex vil si at kommunikasjon kan gå begge retninger samtidig (in og ut). Dette stemmer bra overens med når en snakker med andre over mobiltelefon eller VoIP tjenester.

Half-duplex vil si at kommunikasjon bare kan gå en vei av gangen. Dette kan sammenlignes med walkie-talkie.

Man kan kjøra fleire linjer parallelt eg. PCIe x1, x4, x8, x16. Ein kan setta eit x1 kort i ein slot til x16, men ikkje omvendt. Benytttest no til meir enn bare vanlige tilleggs kort. SSD, Ekstern hardware. Er gunstig i servere fordi dei gir rask lagring og GPU

## 2 Linux

UNIX(Uniplexed Information and Computer Systems) blei først laga på slutten av 60-tallet av blant anna MIT, ATandT, GE og var i begynnelsen ein lukka kilde kode. Da finns også åpne UNIX system som f.eks. FreeBSD. I 1991 kom Linux, Da blei utvikla av MINIX og vart publisert av LINUX-kjernen. I 83 starta Richard Stallman GNU prosjektet som skulle vær ein UNIX lignande, åpent operativsystem og i 91 hadde dei komt så langt at bare kjernen mangla. Torvalds som utvikla Linux og GNU teamet slo så samen prosjekta og derfor er da korrekta namnet GNU/LINUX.

Opensource eller åpen kildekode betyr at koden er:

- Gratis å videredistribuere programmet
- Programmet kan brukes av kven som helst til kva som helst
- Kildekode er tilgjengelig og kjent
- Tillatt å gjøre egne endringer i programmet

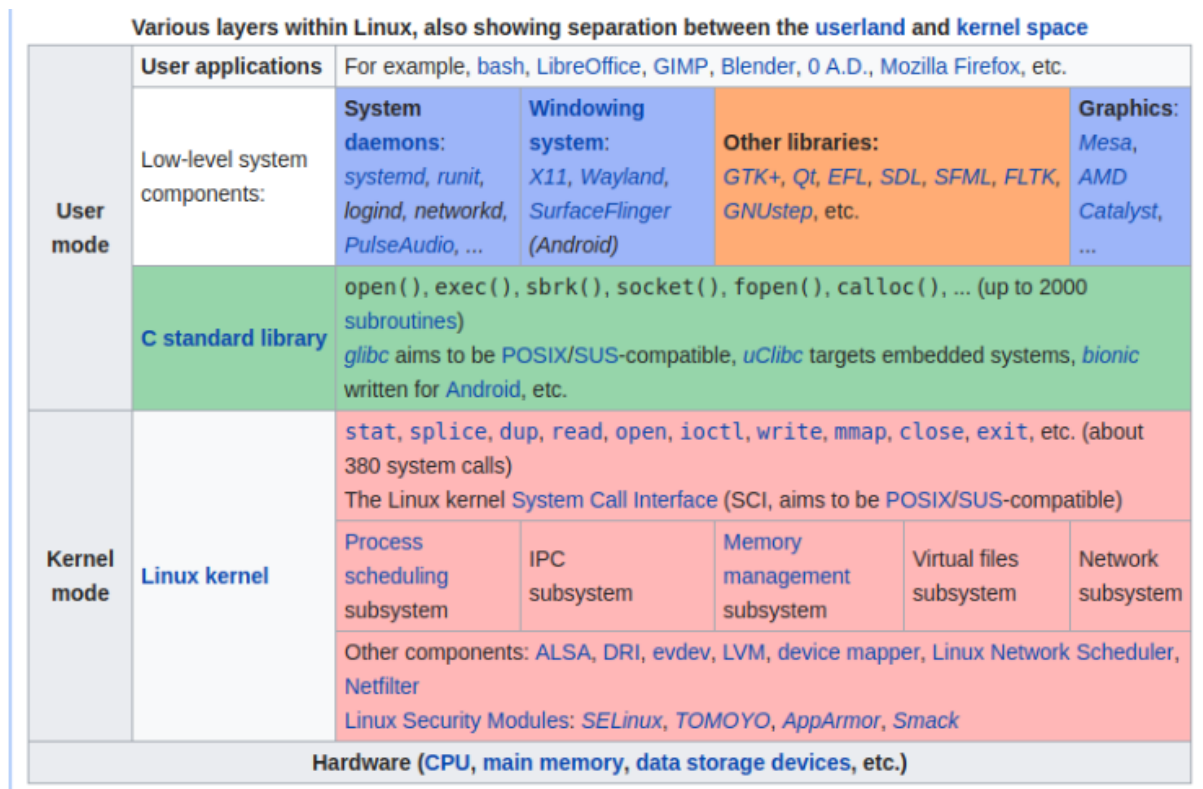
nokre kjente Linux OS:

- RedHat - Fedora
- CentOS
- Debian
- Slackware
- Gentoo
- Knoppix
- Ubuntu
- Suse

### 2.1 Oppbygging

Linux kjernen er monolittisk oppbygd. Da vil seie at heile OS er implementert i kjernen. Alle funksjonane som er implementert blir lasta opp ved oppstart. Linux har støtte for mange modular, og desse blir lasta etter behov. Da som er fint med Linux er at det er hotswap og ein treng ikkje å starta på nytt for å lasta opp modulane og bruka dei.

Moduler kan benyttas for komponenta som oppdateres ofta ein slepp å oppdatere kernel og når ein lastar ned modulane så slepp ein å starta på nytt ein kan utvikla modulane uten å utvikla kjernen Moduler kan benyttas for komponenta som blir bruk sjeldent Last modul kun ved behov. Tar ikkje opp plass i minne



Figur 1: Kjernen

## 2.2 Katalogtre

I UNIX system er alle enheter satt inn i eit filtre der topp katalogen blir kalla root. Alle filer og enheter på maskina blir plassert inni detta treet. Også filsystem frå andre maskiner må monterast ("mountes"). Forskjellige deler av treet kan godt ligna på forskjellige partisjonar

### 2.2.1 Filsystem

Filsystemet bestem korleis filene skal blir lagret. Dette kan påvrika hastigheten og påliteligheten. Da er ein mulighet å ha fleire typer filsystem på ei maskin. ext2 vart lenge brukt som standard, men Journaliserte filsystem er no meir vanleg

-Lagrer metadata om disk slik at data ikke så lett går tapt ved f.eks. strømbrudd

-ext3 har vært mykje brukt

-ext4 er bakoverkompatibel med ext2 og ext3 (TRIM, journaling gjør det mindre egnet for SSD)

-Volum på 1 exabyte (10<sup>18</sup>) og filer på 16 terabyte (10<sup>12</sup>)

-Btrfs(SSD, ikke journal, TRIM), Reiser4 og JFS (IBMs versjon) er også open source

Kan også lesa og skriva FAT, FAT32, NTFS og HFS

ext = extended, ext4 max volum = 1 exbibyte 2<sup>60</sup> byte, max file size 16 tebibyte

XFS = Spesialisert for store "data chunks" 8 exbibyte, max file size 8 exibyte

BTRFS = B-Tree FS, (Butter / Better FS) 16 exbibyte, max file size 16 exibyte

F2FS (Flash-Friendly File System) = Nytt, laget av samsung for å støtte den nye typend NAND disk.

max vol 16TB, max file size 3.94TB

JFS (Proprietært, gammelt og ikke mye brukt)

### 2.2.2 Brukere/grupper

Linux er i grunn eit fleirbruks system. Der root er administratoren og har tilgang til alt. Brukere og grupper må bli oppretta og tildelt av root. Fleire brukere kan vær medlem av ein eller fleire grupper. I gruppene har alle som er medlem av gruppa rettighetene til filene.

I nyere distroer så har ikkje root lenge passord, men ein kan fortsatt bruke root ved hjelp av sudo

kommanden.

### 2.2.3 Rettigheter

Rettigheter settes på filer og mapper Kategoriene er U G O D = Directory U = User (bruker) G = Group (gruppe) O = Other (andre) Rettighetene er r = read, w = write og x = execute Rettigheter kan settes med chmod nnn directory/files Eier kan endres med chown (user:group) directory/files Execute på mappe betyr å liste (0 - noPermission, 1 execute, 2 write, 4 read) Kan settes sammen for å kombinere tillatelser 111 = alle får execute, 222 = alle får write, 444 = alle får read, 333 = alle får execute og write, 555 = alle får read and execute, 666 = alle får write og read, 777 = alle får execute, write og read.

### 2.2.4 Kommands

Hjelpfunksjoner

Manual: man <program>

help, info, (kan også sjekke /usr/share/doc)

Hurtigtaster

tab, ctrl+A, ctrl+E, ctrl+Left, ctrl+Right, ctrl+U, ctrl+K, ctrl+C, up, down

Bevege seg i katalogtreet

ls, cd, pwd

Slette, kopiere og flytte

cp, rm, mkdir, rmdir, dd, mv

Printing av informasjon

cat, less, more

Søk etter filer

find

Sette rettigheter

chmod, chown, chgrp

Opprette brukere og administrere brukere (bare root, evt med sudo)

adduser, addgroup, passwd, usermod

Installering og administrasjon av programmer

apt (yum på Red Hat og derivater av den)

Redigere filer

nano, vim

Håndtering av katalogtreet

df, du, mount, dmesg

Konfigurering av programvare

/etc, man, service, systemctl

Håndtere prosesser ps, &, kill



Standard oppbygging av kommandoer  
kommando valg/parameter argument

Styrken ligger i at man kan sette mange parametre/brytere på kommandoene  
Ikke alle følger denne oppbyggingen, og noen har standard-argumenter og valg satt

Eksempler på parametere  
Bokstav eller tekst:  
-[bokstav] (eksempel: ls -l)  
-[tekst] (eksempel: ls -help)

Omdirigering av in/ut-kilde < > (Endrer standard inn og ut enhet)  
ls > filer.txt

Pipe: |  
kan sende resultat av en kommando som input til en annen  
ls | less  
cat navneliste.txt | grep knut

Jokertegn/wildcards  
?: ett tegn  
\*: Alle tegn  
ls \*.txt

Root-permissions  
sudo <- Samme som "kjør som administrator" i window  
sudo !! <- setter sudo foran siste kommando

## 2.3 SSH

SSH eller Secure SHell er ein kommand som ofte blir brukt til å få tilgang til servere. SSH er kryptert fra ende til ende så er ganske sikker. Bruker sertifikater for autentisering av server og bruker (valgfritt) Innlogging med de eksisterende brukerne på systemet Hovedsakelig dette som brukes i dag

Autentisering brukes av klienten for å bekrefte at den kobler til riktig server. Klienten kan stoppe innlogging før den har sendt noe informasjon til serveren. Forhindrer tyveri av brukernavn/passord ved å sette opp en falsk server. Sertifikater kan signeres med en kjent nøkkel og dermed verifiseres. Ved første pålogging tar klienten vare på public key'en til serveren (lagres som known host). Ved videre pålogginger blir serveren autentisert mot den lagrede public key'en. Autentisering kan brukes av serveren for å verifisere brukeren. Nøkler brukes for å autentisere og logge inn i stedet for brukernavn / passord. Nøkkelen kan krypteres lokalt med ett passord for å sikre den. Krever at public keyen til brukeren er skrevet til `/.ssh/authorized_keys`. Ikke obligatorisk – brukernavn / passord brukes hvis public key'en ikke er kjent. Du kan installere SSH key'en din automatisk med `ssh-copy-id`. `ssh-copy-id host.domain.com`.

SSH kan også brukes for sikker overføring av filer til/fra server. SFTP – Oppfører seg som vanlig FTP, men sikker. FileZilla er ein gratis FTP klient som også støtte SFTP filezilla.  
Adresse er `sftp://<serveradresse>:<port(std:22)>`.

SSH brukernavn/passord/nøkler. SCP – Secure Copy. Kopier filer over SSH fra kommandolinje til/fra server. `scp <fra> <til>`. `scp enfil.txt brukernavn@server:/home/brukernavn/dokumenter`. Kopierer filen enfil.txt fra nåværende mappe til mappe spesifisert på server.  
`scp brukernavn@server:/home/brukernavn/dokumenter/enfil.txt .` Kopier fil fra server til nåværende mappe (Legg merke til punktumet). `-r` kan brukes for å kopiere hele kataloger.

`ssh/config` er en global konfigurasjonsfil for SSH. Du kan legge inn konfigurasjonen for servere du bruker ofte her. Da slipper du taste inn brukernavn og adresse hver gang du skal koble til. Du kan også sette opp port forwards. En port forward gir deg tilgang til tjenester gjennom SSH.

## 2.4 Enheter

Enheter Alle enheter (interne og eksterne) er representert i filsystemet i Linux

/dev/sd? - Harddisker

/dev/input/\* - mus o.l.

/dev/md? - RAID-enheter

/dev/mapper/\* - Virtuelle blokkenheter

/dev/disk-by

Partisjoner Blokkenheter (lagring) deles opp i partisjoner En blokkenhet kan ha 0 eller flere partisjoner Disse nummereres, hvor 1 er den første, f. eks: /dev/sda1 /dev/sdc1 /dev/sdc2 Det er partisjonene som har filsystemer på seg Partisjoner kan lages og fjernes med verktøyet fdisk Obs: Fjerning av partisjon gjør alle filer/data på partisjonen utilgjengelig

### 2.4.1 Loop-enheter

Virtuelle enheter basert på filer. Lager en "fake" harddisk som egentlig bare er en fil. Kan brukes til å lage midlertidige test-enheter. Kan også brukes for å montere opp harddisk-images laget med dd. Størrelsen på loop-enheten blir størrelsen på filen som benyttes

Oppsett Lage fil med spesifikk størrelse (100MiB):

```
dd if=/dev/zero of=testfil.bin bs=1M count=100
# Administreres med losetup
# Sette opp en loop-enhet basert på testfil.bin:
losetup /dev/loop0 testfil.bin
# Fjerne loop-enhet:
losetup -d /dev/loop0
```

## 2.5 Raid

Redundant Array of Independent Disks eller RAID er å slå sammen flere blokkenheter (typisk harddisker) til en enhet. Dette gjøres for å øke størrelse, hastighet og/eller pålitelighet.

Raidnivåer

Ulike raid-nivåer har ulike egenskaper – hastighet og/eller pålitelighet Hva man velger er avhengig av behov/bruksmønster

Oppsett og bruk

RAID kan implementeres både i software og hardware

Software: Operativsystemer ser alle harddiskene og håndterer RAID selv

Hardware: En egen kontroller (chip) styrer RAID, operativsystemet ser ikke diskene

Filsystemer kan legges rett på en RAID-enhet, eller på en partisjon

Nivå	Metode	Lese	Skrive	Pålitelighet	Lagringsplass
RAID 0	Striping	nX	nX	Kan miste 0	Alt
RAID 1	Speiling	nX	1X	Kan miste n-1	1 disk
RAID 5	Paritet	nX-	nX-	Kan miste 1	n - 1
RAID 6	Paritet	nX-	nX-	Kan miste 2	n - 2
RAID 10	Stripe/Speil	StxSp	StxSp	Inntil siste disk i et speil	n speil

Figur 2: Raid typer

### 2.5.1 Software raid

Administreres med verktøyet mdadm - Multiple Device Administrator  
RAID kan lages på toppen av harddisker, partisjoner eller virtuelle enheter  
Støtter mange RAID-nivåer – se man mdadm

#### Oppsett

Lag en RAID1-enhet med /dev/sdg og /dev/sdh:  
mdadm –create /dev/md0 –level=mirror –raid-devices=2 /dev/sdg /dev/sdh  
RAID-enheter får en sti som alle andre enheter - /dev/md0  
Man kan så lage filsystem på denne enheten med mkfs  
Arrayet settes sammen (assemble) automatisk ved oppstart

#### Overvåkning

Enkel status står i /proc/mdstat:  
cat /proc/mdstat  
Detaljert status:  
mdadm –detail /dev/md0

#### Håndtering av feil

Støtter aktive reservedisker  
Disker som er til overs blir active spare  
Reservedisker i arrayen benyttes automatisk ved feil  
Støtter hotswap – bytt disk mens filsystemet er i bruk  
Disken må først taes ut av RAID med mdadm, så legges en ny til  
Rebygging kan ta mange timer – avhengig av størrelse på diskene

#### Portabilitet

Hele arrayen kan flyttes til en annen maskin enkelt ved å flytte diskene  
Så lenge mdadm er installert vil arrayet komme opp  
Filsystemet kan så monteres for å hente ut nødvendige data

#### Filsystemer

RAID-enheter kan partisjoneres eller brukes direkte  
Et filsystem kan dermed legges rett på /dev/md? hvis ønskelig  
Arrayen må være satt sammen før filsystem kan lages / monteres  
Arrays settes automatisk sammen ved oppstart  
Hvis alle diskene er tilgjengelige  
Hvis alle diskene har status OK

#### Behov for mer diskplass?

Støtter grow – Legg til en ny disk for å øke størrelsen  
Først må disken legges til arrayen med mdadm som spare  
Så økes størrelsen med mdadm  
Filsystemet er tilgjengelig mens dette foregår - ingen nedetid

#### mailutils

mailutils er en pakke som gjør det mulig å sende e-post lokalt på systemet  
Dette kan brukes av andre verktøy for å varsle om feil  
Ved installering av mailutils så velger dere local only  
Det er også mulig å sette opp slik at den sender e-post eksternt

#### Overvåkning av RAID arrayer

md bruker mailutils automatisk til å varsle om feil hvis den er installert  
Mottaker kan konfigureres i filen /etc/mdadm/mdadm.conf

Standard-mottaker er root-brukeren

smartmontools

Pakke med verktøy for å overvåke harddisker

Vise feilinformasjon om harddisk:

`smartctl -a /dev/sda`

Smartmontools bruker mailutils automatisk til å varsle om feil

Mottaker kan konfigureres i filen `/etc/smartd.conf`

Standard mottaker er den første ordinære brukeren på systemet

## 2.6 Scripting

Definisjon

Skripting er en underklasse av koding

Et skript blir ikke compilert

Ofte kode for å automatisere repeterende oppgaver

Fordeler

Portabelt

Sparer tid

Uniformt

Bakdeler

Ytelse

Må ha "Interpreter"

Bourne Again SHell (macOS pre Cat., Cygwin, MinGW, Android ++)

Nesten helt kompatibelt med SH (Bourne Shell)

- Har "kompatibilitetsmodus" for økt kompatibilitet.

Unix bryr seg i utgangspunktet ikke om "file extensions"

"Shebang" fortell operativsystemet hvordan det skal tolke fila

NB! Husk å gjøre file "executable" i linux

Gjøres med "chmod +x filnavn"

Kan verifiseres med `ls -l`, sjekk at fila har en x i user permissions

Using the interpreter

Using the interpreter

`#!/bin/sh`

`#!/bin/bash`

`#!/bin/python`

← Bourne Shell (POSIX Standard)

← Bash (BourneAgain) Shell (Common)

← Python Skript

Using the program search path

`#!/usr/bin/env sh`

`#!/usr/bin/env bash`

`#!/usr/bin/env python`

← Bourn Skript

← Bash Skript

← Python Skript

### 2.6.1 Variabler

Kan kun inneholde tall, bokstaver og underscore

Kan ikke begynne med tall

Standarden for globale variabler og ENV variabler i Unix shell er UPERCASE  
 Dine/lokale variabler skal være lowercase  
 Dette for å lettere kunne skille i koden  
 \_VAR og \_var indikerer en statisk verdi og skal ikke endres.

BASH Variabler:

Gyldig UPPER og lower:

```
_ALI
TOKEN_A
VAR_1
VAR_2
```

Ikke Gyldig:

```
2_VAR
-VARIABLE
VAR1-VAR2
VAR_A!
```

BASH Variabel Deklarasjon:

Gyldig:

```
VAR_A="value"
_VAR_B="value"
var="value"
_var="value"
```

Ikke Gyldig:

```
VAR_A = "value"
VAR_A= "value"
VAR_A = "value"
```

Vi må være OBS på spaces i bash

```
[-z '$VAR'] er ikke greit

./if_statements: line 15: [-z: command not found

[ -z '$VAR' ] er greit
```

Ofte de enkle feilene som feller oss, så derfor tester vi koden ofte  
 IDEen kan hjelpe oss mye her

## 3 Nettverk

### 3.0.1 Standar

In general, network standards refer to a series of definitions for data (and Voice) networks that are published by various organizations such as IEEE, CCITT (ITU), EIA/TIA and ISO. These standards define how networks are supposed to work in regards to architecture, electrical, interface and overall process standards.

Protocols are data communication rules designed that specify the order and sequence of the bits in a data stream. Every time you access the Internet, send an email, print a document, etc your computer is using data protocols to sent that info to the right place in the right form.

Both standards and protocols are important so that people can design equipment and software that can inter-operate with other equipment.