12 Linux kommandoer, der skal bruges til systemhardwarekontrol

12 Linux kommandoer, der skal bruges til systemhardwarekontrol til at få indsigt i dit system og hardware som er afgørende for optimering og fejlfinding. Disse nøglekommandoer afslører detaljerede oplysninger om din computers indre funktioner.

I denne guide vil vi se på 12 essentielle kommandoer, som enhver Linux bruger bør kende, uanset om de er en erfaren sysadmin eller en nysgerrig begynder.

Disse kommandoer giver et indblik i din Linux maskines indre funktioner. Jeg vil dække en række kommandoer, der giver indsigt i forskellige aspekter af dit system, fra forståelse af dit systems arkitektur til overvågning af systemets ydeevne.

Dette handler ikke kun om at udføre kommandoer; det handler om at udvikle et forhold til dit Linux system, og lære at kommunikere med det mere effektivt.

Hver kommando fortæller en unik historie om dit system, og at forstå disse kommandoer gør dig ikke bare til en bruger, men en kender af Linux operativsystemet.

Så lad os tage fat i vores tastaturer og begynde vores udforskning af kommandolinjeuniverset, og opdage værktøjer og kommandoer, der vil forbedre din færdighed og tillid til at håndtere og forstå dit Linuxsystem.

12 vigtige kommandoer til at kontrollere system og hardwareoplysninger

1. uname – Grundlæggende systemoplysninger

Syntaks: **`uname -a`**

Eksempel på output:

```
carl@andersen in ~ as 🧙 took 4ms
λ uname -a
Linux andersen 6.6.1-zen1-1-zen #1 ZEN SMP PREEMPT_DYNAMIC Wed, 08 Nov 2023 16:05:16 +0000 x86_64 GNU/Linux
```

Denne kommando giver et hurtigt overblik over dit system, inklusive kerneversionen, værtsnavnet og hardwarearkitekturen. Det er utrolig nyttigt til at få et overblik over det system, du arbejder på, på højt niveau, især når du diagnosticerer kompatibilitetsproblemer.

2. lsb_release – Oplysning af distributionsspecifik information

Syntaks: **lsb_release -a**

Eksempel på output:

```
carl@andersen in ~ took 5ms
λ lsb_release -a
LSB Version:    n/a
Distributor ID: Garuda
Description:    Garuda Linux
Release:        Soaring
Codename:       Spizaetus
```

Kommandoen **lsb_release** er specifik for Linux distributioner og giver detaljerede oplysninger om den distribution, du bruger. Dette er afgørende, når du administrerer softwareafhængigheder, der kan variere mellem distributioner.

3. hostnameectl – Systemidentifikation

Syntaks: **hostnamectl**
Eksempel på output:

```
┌carl@andersen in ~ took 4ms
└λ hostnamectl
 Static hostname: andersen
       Icon name: computer-desktop
         Chassis: desktop 🖥️
      Machine ID: fa448179c2d84fdc8c54fad1185844da
         Boot ID: a0cc60ea2e9d47beb0f587f871cec9a8
Operating System: Garuda Linux
          Kernel: Linux 6.6.1-zen1-1-zen
    Architecture: x86-64
 Hardware Vendor: Gigabyte Technology Co., Ltd.
  Hardware Model: B660M DS3H DDR4
Firmware Version: F20
   Firmware Date: Tue 2022-10-25
    Firmware Age: 1y 2w 6d
```

Hostnamectl er især nyttigt for systemadministratorer og dem, der administrerer flere maskiner, da det giver detaljerede oplysninger om systemet, herunder værtsnavn, operativsystem, kerne og hardwaredetaljer.

4. lscpu – Processordetaljer

```
carl@andersen in ~ took 4ms
λ  lscpu
Arkitektur:                x86_64
  Op-tilstande for CPU:    32-bit, 64-bit
  Adressestørrelser:       46 bits physical, 48 bits virtual
  Byterækkefølge:          Little Endian
CPU'er:                    20
  Tilkoblet cpu(er) liste: 0-19
Leverandør-id:             GenuineIntel
  Modelnavn:               12th Gen Intel(R) Core(TM) i7-12700
    CPU-familie:           6
    Model:                 151
    Tråde per kerne:       2
    Kerner per sokkel:     12
    Sokler:                1
    Modelserie:            2
    CPU(s) scaling MHz:    57%
    CPU maks. MHz:         4900,0000
    CPU min. MHz:          800,0000
    BogoMIPS:              4224,00
    Flag:                  fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts a
                           cpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art arch
                           _perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf tsc_known_freq
                            pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm ss
                           e4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm
                           abm 3dnowprefetch cpuid_fault ssbd ibrs ibpb stibp ibrs_enhanced tpr_shadow flexprior
                           ity ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid rdseed adx s
                           map clflushopt clwb intel_pt sha_ni xsaveopt xsavec xgetbv1 xsaves split_lock_detect
                           user_shstk avx_vnni dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp hwp
                           _pkg_req hfi vnmi umip pku ospke waitpkg gfni vaes vpclmulqdq tme rdpid movdiri movdi
                           r64b fsrm md_clear serialize pconfig arch_lbr flush_l1d arch_capabilities
Virtualization features:
  Virtualisation:          VT-x
Caches (sum of all):
  L1d:                     512 KiB (12 instances)
  L1i:                     512 KiB (12 instances)
  L2:                      12 MiB (9 instances)
  L3:                      25 MiB (1 instance)
NUMA:
  NUMA-knuder:             1
  NUMA-knuder0 CPU'er:     0-19
Vulnerabilities:
  Gather data sampling:    Not affected
  Itlb multihit:           Not affected
  L1tf:                    Not affected
  Mds:                     Not affected
  Meltdown:                Not affected
  Mmio stale data:         Not affected
  Retbleed:                Not affected
  Spec rstack overflow:    Not affected
  Spec store bypass:       Mitigation; Speculative Store Bypass disabled via prctl
  Spectre v1:              Mitigation; usercopy/swapgs barriers and __user pointer sanitization
  Spectre v2:              Mitigation; Enhanced / Automatic IBRS, IBPB conditional, RSB filling, PBRSB-eIBRS SW
                           sequence
```

Lscpu viser detaljerede oplysninger om CPU-arkitekturen, herunder antallet af kerner, tråde, CPU-familien og aktuelle driftsfrekvenser. Det er vigtigt for at forstå dit systems behandlingsmuligheder, især når du optimerer ydeevnen eller vurderer, om dit system kan håndtere bestemte applikationer

Forklaring:

Flag er en måde at indstille indstillinger og sende argumenter til de kommandoer, du kører. Kommandoer du kører vil ændre deres adfærd baseret på hvilke flag der er sat. Du bør læse dokumentationen til hver kommando for at vide, hvilke flag der er tilgængelige.

For eksempel vil kørsel af **ls** med flaget **-l** (**ls -l**) inkludere flere oplysninger i resultatet og ændre formatet på det, der returneres.

```
carl@andersen in ~ took 6ms
λ ls -l
drwxr-xr-x    - carl  9 nov 12:59 .bluefish
drwx------    - carl 14 nov 11:45 .cache
drwxr-xr-x    - carl  1 nov 09:25 .cargo
drwxr-xr-x    - carl 29 okt 09:22 .cinnamon
drwxr-xr-x    - carl 14 nov 11:05 .config
drwx------    - carl  2 nov 10:11 .fltk
drwxr-xr-x    - carl 29 okt 09:22 .icons
drwxr-xr-x    - carl 30 okt 14:01 .local
drwx------    - carl 30 okt 14:13 .mozilla
drwx------    - carl  8 nov 12:44 .newsboat
drwxr-xr-x    - carl 14 nov 11:05 .openshot_qt
drwx------    - carl 30 okt 14:51 .pki
drwx------    - carl  5 nov 08:21 .putty
drwxrwx---    - carl 31 okt 18:01 .sane
drwxr-xr-x    - carl 12 nov 15:25 .ssr
drwxr-xr-x    - carl 31 okt 06:17 .themes
drwx------    - carl 30 okt 15:12 .thunderbird
drwxr-xr-x    - carl 14 nov 11:44 Billeder
drwxrwxr-x    - carl 14 nov 10:59 Dokumenter
drwxr-xr-x    - carl 10 nov 17:50 Hentet
drwxr-xr-x    - carl 30 okt 14:19 MEGA
drwxr-xr-x    - carl 30 okt 14:01 Musik
drwxr-xr-x    - carl 30 okt 14:01 Offentligt
drwxr-xr-x    - carl 30 okt 14:01 Skabeloner
drwxr-xr-x    - carl 14 nov 09:55 Skrivebord
drwxr-xr-x    - carl 12 nov 09:53 Videoklip
drwxr-xr-x    - carl 12 nov 09:43 'VirtualBox VMs'
.rw------- 5,3k carl 14 nov 11:44 .bash_history
.rw-r--r--   21 carl 21 maj 12:56 .bash_logout
.rw-r--r--   57 carl 21 maj 12:56 .bash_profile
.rw-r--r-- 3,1k carl 16 okt 18:04 .bashrc
.rw-r--r--   27 carl 30 okt 14:01 .dmrc
.rw-r--r--  245 carl  4 nov 09:13 .gtkrc-2.0
.rw-r--r--  189 carl  2 okt 16:05 .profile
.rw-------   53 carl 14 nov 10:57 .Xauthority
.rw-r--r-- 1,5k carl  2 okt 16:05 .xinitrc
.rw-r--r-- 1,5k carl  2 okt 16:05 .xinitrc
.rw-------  64k carl 14 nov 12:01 .xsession-errors
.rw------- 712k carl 14 nov 09:55 .xsession-errors.old
.rw-r--r-- 4,9M carl 31 okt 17:15 1965104.ppt
.rw-r--r-- 7,0M carl 31 okt 17:12 6999848.ppt
.rw-r--r-- 2,5G carl 12 nov 13:30 garuda-xfce-linux-lts-231029.iso
.rw-r--r--   99 carl 12 nov 13:30 'garuda-xfce-linux-lts-231029.iso (1).sha256'
.rw-r--r--   99 carl 12 nov 09:22 garuda-xfce-linux-lts-231029.iso.sha256
.rw-r--r-- 560k carl 25 dec  2021 NessYawl101bDigLevel.jpg
```

Virtualization features: Linux virtualisering refererer til en proces, hvor mere end én virtuel maskine kan installeres i Linux. Du kan installere Windows, Mac, FreeBSD, Solaris, IBM OS/2 og selvfølgelig Linux i en virtuel maskine. virtualisering skal slås til i Bios

Vulnerabilities:

Sårbarheder: Som enhvert operativsystem har Linux ikke immunt over for sårbarheder.

Linux systemer bruger ofte en bred vifte af software og applikationer, og disse kan være sårbare over for udnyttelser og andre sikkerhedssvagheder. For eksempel kan sårbarheder i webbrowsere, e-mail-klienter og andre almindelige applikationer udnyttes af angribere til at få adgang til et system eller stjæle følsomme data. For at mindske denne risiko:

Hold software og applikationer opdateret: Sørg for regelmæssigt at anvende opdateringer og patches for at rette kendte sårbarheder.

Brug velrenommeret software og applikationer:
top – Real-time system monitor
Vælg software og applikationer fra velrenommerede kilder, og undgå at downloade software fra upålidelige websteder.

Brug antivirussoftware: Antivirussoftware kan hjælpe med at opdage og fjerne malware, der kan udnytte sårbarheder i software og applikationer.

5. free – Hukommelsesbrug i MB

Syntaks: **free**

Eksempel på output:

```
┌carl@andersen in ~ took 5ms
└λ free
              total        used        free      shared  buff/cache   available
Mem:      131672988     5453348   122172840      936240     6096240   126219640
Swap:     131672060           0   131672060
```

Hukommelsesbrug i GB

Syntaks: **free -h**

free kommandoen, der bruges her med flaget **-h** viser hukommelsen i gigabyte, giver et øjeblikkeligt overblik over systemets hukommelsesforbrug, inklusive total, brugt og ledig hukommelse. Det er især nyttigt til at overvåge dit systems hukommelsesydelse under forskellige belastninger

Eksempel på output:

```
carl@andersen in ~ took 5ms
λ free -h
              total        used        free      shared  buff/cache   available
Mem:          125Gi       5,3Gi       116Gi       919Mi       5,8Gi       120Gi
Swap:         125Gi          0B       125Gi
```

free kommandoen, der bruges her med flaget -h viser hukommelsen i gigabyte, giver et øjeblikkeligt overblik over systemets hukommelsesforbrug, inklusive total, brugt og ledig hukommelse. Det er især nyttigt til at overvåge dit systems hukommelsesydelse under forskellige belastninger

6. df – Diskplads

Syntaks: **df -h**

Eksempel på output:

```
┌─carl@andersen in ~ took 6ms
└─λ df -h
Filsystem      Størr Brugt  Tilb Brug% Monteret på
/dev/nvme0n1p2  932G  110G  818G   12% /
devtmpfs        4,0M     0  4,0M    0% /dev
tmpfs            63G  135M   63G    1% /dev/shm
efivarfs        256K  124K  128K   50% /sys/firmware/efi/efivars
tmpfs            26G  2,0M   26G    1% /run
/dev/nvme0n1p2  932G  110G  818G   12% /home
/dev/nvme0n1p2  932G  110G  818G   12% /root
/dev/nvme0n1p2  932G  110G  818G   12% /srv
/dev/nvme0n1p2  932G  110G  818G   12% /var/cache
/dev/nvme0n1p2  932G  110G  818G   12% /var/log
/dev/nvme0n1p2  932G  110G  818G   12% /var/tmp
/dev/nvme0n1p1  300M  576K  299M    1% /boot/efi
tmpfs            63G  7,7M   63G    1% /tmp
tmpfs            13G  256M   13G    2% /run/user/1000
```

df med flaget **-h** viser mængden af diskplads, der bruges og er tilgængelig på dine filsystemer.

7. `lsblk` – blokenheder og partitioner

Syntaks: **`lsblk`**

Eksempel på output:

```
 ┌─carl@andersen in ~ took 6ms
 └─λ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda           8:0    0   1,8T  0 disk
└─sda1        8:1    0   1,8T  0 part
sdb           8:16   1  28,7G  0 disk
├─sdb1        8:17   1   2,4G  0 part /run/media/carl/GARUDA_XFCE_RAPTOR
└─sdb2        8:18   1     4M  0 part
zram0       254:0    0 125,6G  0 disk [SWAP]
nvme0n1     259:0    0 931,5G  0 disk
├─nvme0n1p1 259:1    0   300M  0 part /boot/efi
└─nvme0n1p2 259:2    0 931,2G  0 part /var/tmp
                                      /var/log
                                      /var/cache
                                      /srv
                                      /root
                                      /home
                                      /
```

Lsblk er en utrolig nyttig kommando til at få et klart overblik over alle blokenheder (som harddiske og SSD'er), der er tilsluttet dit system, sammen med deres monteringspunkter.

Hvad er forskellen mellem lsblk og DF? "df" viser diskbrug og filsystemoplysninger. "lsblk" viser blokenheder og partitioner.

8. dmidecode – Skjulte hardwaredetaljer - skal installeres

Syntaks: **sudo dmidecode -t system**

Eksempel på output:

```
 ┌carl@andersen in ~ took 5ms
 └λ sudo dmidecode -t system
[sudo] adgangskode for carl:
# dmidecode 3.5
Getting SMBIOS data from sysfs.
SMBIOS 3.5.0 present.

Handle 0x0001, DMI type 1, 27 bytes
System Information
        Manufacturer: Gigabyte Technology Co., Ltd.
        Product Name: B660M DS3H DDR4
        Version: Default string
        Serial Number: Default string
        UUID: 03560274-043c-0522-9206-ae0700080009
        Wake-up Type: Power Switch
        SKU Number: Default string
        Family: B660 MB

Handle 0x0023, DMI type 12, 5 bytes
System Configuration Options
        Option 1: Default string

Handle 0x0024, DMI type 32, 20 bytes
System Boot Information
        Status: No errors detected
```

Dmidecode er et kraftfuldt værktøj til at udtrække hardwareinformation fra systemets BIOS eller firmware, ofte mere detaljeret end andre kommandoer kan give. Det kræver administrative rettigheder og er fantastisk til at få specifikke hardwaredetaljer som producent, produktnavn og serienummer.

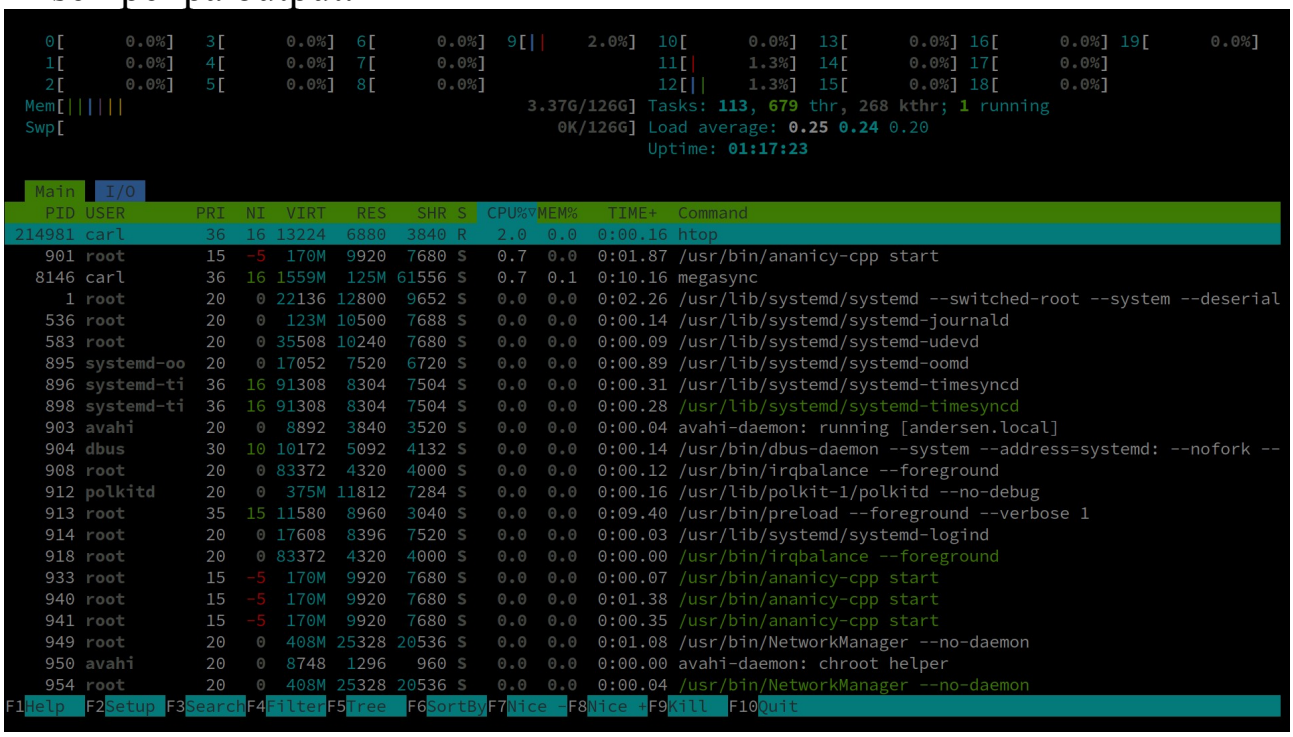
Video: https://youtu.be/VhljjghqAT8

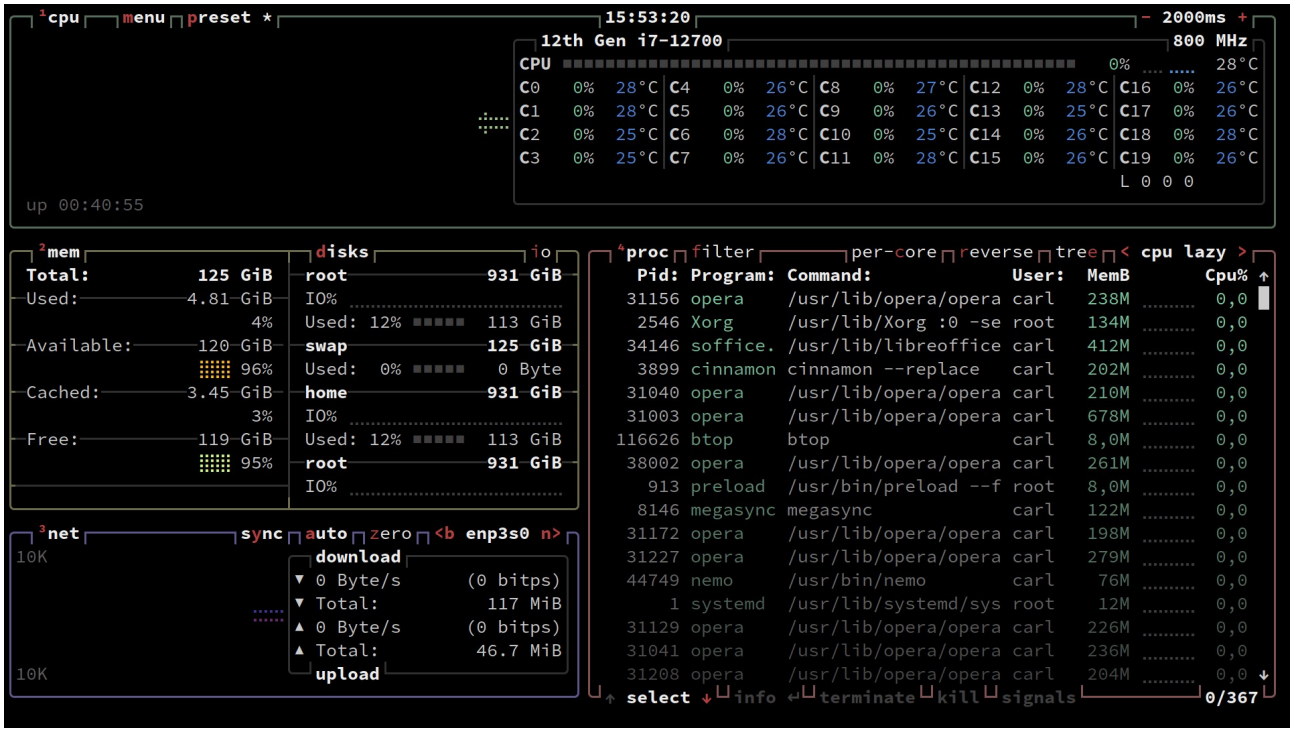
Hjemmeside: https://www.nongnu.org/dmidecode/

9. htop – btop Systemmonitor

Syntaks: **htop eller btop**

Eksempel på output:





Btop++ er et kommandolinjeværktøj på tværs af platforme, og det kan køre på Linux, Windows, BSD og MacOS.

Btop++ kommer med en håndfuld layout forudindstillinger, som du kan bruge til hurtigt at ændre udseendet og følelsen af det.
Der er understøttelse for mus, så du kan navigere med musen. Du kan konfigurere dens forskellige muligheder ved hjælp af en indbygget præference menu, der fungerer i selve terminalen. Det inkluderer en mulighed for at vise både opsummerede og detaljerede statistikker. Andre hovedfunktioner i Btop++ inkluderer dets evne til at sortere og filtrere processer, et indbygget trævisningslayout, grafer og plots, der viser overtidsforbrug af ressourcer, batteriindikator, farvet output, vim stil tastaturgenveje, tilpasselige temaer og et indbygget ur.

Video: https://youtu.be/OUZJ8rWiiug

11. ip addr – Netværksgrænseflader

Syntaks: **ip addr**

Eksempel på output:

```
┌─carl@andersen in ~ took 5ms
└─λ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 74:                brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global dynamic noprefixroute enp3s0
       valid_lft 82807sec preferred_lft 82807sec
    inet6 fe80::                    /64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Kommandoen ip addr er et vigtigt værktøj for netværksadministratorer og alle, der har brug for at fejlfinde eller konfigurere netværksgrænseflader. Den viser detaljerede oplysninger om alle netværksgrænseflader på dit system, inklusive loopback-grænseflader, Ethernet, Wi-Fi og andre netværksadaptere. Outputtet viser hver grænseflades navn, tilstand (op/ned), MAC-adresse, IP-adresse og andre relevante data.

12. fastfetch – Systemoplysninger

Syntaks: **fastfetch**

Eksempel på output:

```
┌─carl@andersen in ~ took 6ms
└─λ fastfetch
                    .%;888:8898898:                carl@andersen
                  x;XxXB%89b8:b8%b88:              -------------
              .8Xxd                 8X:.           OS: Garuda Linux x86_64
            .8Xx;                     8x:.         Host: B660M DS3H DDR4
          .tt8x           .d             x88;      Kernel: 6.6.1-zen1-1-zen
       .@8x8;           .db:               xx@;    Uptime: 1 hour, 7 mins
     ,tSXX°           .bbbbbbbbbbbbbbbbbbbB8x@;     Packages: 1597 (pacman)[stable]
   .SXxx            bBBBBBBBBBBBBBBBBBBBbSBX8;      Shell: bash 5.2.15
 ,888S                                       pd!   Display (ASUS VP248): 1920x1080 @ 60Hz
8X88/                                         q    DE: Cinnamon 5.8.4
8X88/                                              WM: Muffin (X11)
GBB.                                               WM Theme: Sweet-Dark (Sweet-Dark)
 x%88        d888@8@X@X@X88X@@XX@@X@8@X.           Theme: Sweet-Dark [GTK2/3/4]
   dxXd    dB8b8b8B8B08bB88b998888b88x.            Icons: Yaru-olive-dark [GTK2/3/4]
    dxx8o                        .@@;.             Font: Noto Sans (10pt) [GTK2/3/4]
      dx88                     .t@x.               Cursor: Adwaita (24px)
        d:SS@8ba89aa67a853Sxxad.                   Terminal: terminator 3.11.5
          .d988999889889899dd.                     Terminal Font: Source Code Pro (16pt)
                                                   CPU: 12th Gen Intel(R) Core(TM) i7-12700 (20) @ 4,80 GHz
                                                   GPU: Intel AlderLake-S GT1
                                                   Memory: 3,94 GiB / 125,57 GiB (3%)
                                                   Swap: 0 B / 125,57 GiB (0%)
                                                   Disk (/): 109,73 GiB / 931,22 GiB (12%) - btrfs
                                                   Local IP (enp3s0): 192.168.1.100/24 *
                                                   Locale: da_DK.UTF-8
```

Fastfetch er et neofetch-lignende værktøj til at hente systemoplysninger og vise dem på en smuk måde. Det er hovedsageligt skrevet i C, med ydeevne og tilpasningsmuligheder i tankerne. I øjeblikket understøttes Linux, Android, FreeBSD, MacOS og Windows.

Installation: https://github.com/fastfetch-cli/fastfetch