



Laboration – Hårdvara

OBS det kan förekomma små skillnader vad gäller sökvägar samt befintliga kommandon mellan olika distributioner. T.ex. mellan CentOS och Ubuntu,

Material: För att genomföra laborationen behöver man ha tillgång till en dator (vm) med Linux installerat. Windows Subsystem for Linux går att använda också. För vissa övningar kan lokal administratörsbehörighet behövas. I labben utgår det från att man är inloggad med en användare som heter **sysadmin**

Mål: I denna laboration kommer du att utföra följande uppgifter:

- Använda kommandon för att lista hårdvara

Lista och visa hårdvarukonfiguration

I denna uppgift kommer du att köra flera kommandon och undersöka olika filer för att visa din hårdvarukonfiguration.

För att ta reda på vilken typ av CPU du har, kör kommandot **lscpu**:

```
lscpu
```

Ditt utdata bör likna det här:

```
sysadmin@localhost:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                16
On-line CPU(s) list:  0-15
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             16
NUMA node(s):          1
Vendor ID:             AuthenticAMD
CPU family:            25
-More-
sysadmin@localhost:~$
```

Att kunna visa information om din CPU är viktigt när du vill ta reda på om mer avancerade Linux-funktioner kan användas på ditt system. Om du vill ha ännu mer detaljer om dina CPU:er kan du undersöka filen **/proc/cpuinfo**, särskilt "flags" som visar om din CPU har vissa egenskaper.

Använd kommandot **head** med flaggan **-n** för att lista de första 20 raderna i **cpuinfo**-filen:

```
head -n 20 /proc/cpuinfo
```



```
sysadmin@localhost:~$ head -n 20 /proc/cpuinfo
processor       : 0
vendor_id     : AuthenticAMD
cpu family    : 25
model        : 1
model name    : AMD EPYC 7443P 24-Core Processor
stepping     : 1
microcode    : 0xa001173
cpu MHz      : 2844.656
cache size   : 512 KB
physical id  : 0
siblings     : 1
core id      : 0
cpu cores    : 1
-More-
sysadmin@localhost:~$
```

Även om dessa "flags" ligger utanför kursens ram, kan det vara bra att hitta dem när du planerar eller felsöker produktionsystem.

För att ta reda på hur mycket RAM och swap-utrymme som används, använd kommandot `free`:

```
free -m
free -g
```

Utdata visar mängden minne i megabyte när flaggan `-m` används och i gigabyte när `-g` används:

```
sysadmin@localhost:~$ free -m
              total        used         free   shared  buff/cache   available
Mem:          64298       12937         5837        26     45523     50616
Swap:          8191           0         8191
sysadmin@localhost:~$ free -g
              total        used         free   shared  buff/cache   available
Mem:           62          12           5         0         44         49
Swap:           7           0           7
sysadmin@localhost:~$
```

I utdata ovan kan du se att systemet har 64298 megabyte (ungefär 64 gigabyte) fysiskt minne (RAM). Av detta används bara 12937 megabyte, vilket är ett gott tecken på att du har tillräckligt med minne för systemets behov.

Om du skulle få slut på minne används swap. Swap är hårddiskutrymme som tillfälligt används för att lagra data som egentligen ska ligga i RAM.

Minnesvärdena i vår virtuella maskinmiljö kan skilja sig något från värdena i terminalexemplet ovan.

För att se vilka enheter som är anslutna till PCI-bussen kan du använda kommandot `lspci`:

```
lspci
```

Observera att många av enheterna som är anslutna till moderkortet visas i den partiella utdata nedan:



```
sysadmin@localhost:~$ lspci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX
Host bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX
AGP bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev
08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE
(rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev
08)
00:07.7 System peripheral: VMware Virtual Machine Communication
Interface (rev 10)
...
```

Utdata från kommandot `lspci` kan vara till hjälp för att identifiera enheter som inte stöds av Linux-kärnan. Vissa enheter, som grafikkort, kan exempelvis bara ge grundläggande funktionalitet utan att du installerar egenutvecklade drivrutiner. Dock hanterar nya distributioner detta allt snabbare och avancerad hårdvarufunktionalitet blir allt vanligare idag.

Använd kommandot `lspci` med flaggan `-k` för att visa enheter tillsammans med den kärndrivrutin och de moduler som används:

Försök lista de USB-anslutna enheterna:

```
lsusb
```

Så här kan du få information om eventuella USB-minnen du vill montera, samt kringutrustning som mus och tangentbord.

```
sysadmin@localhost:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 001 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
sysadmin@localhost:~$
```

För att hårdvara ska fungera laddar Linux-kärnan vanligtvis en drivrutin eller modul. Använd kommandot `lsmod` för att se vilka moduler som är laddade just nu:

```
lsmod
```

En del av utdata från kommandot visas nedan. Första kolumnen är modulens namn och den andra visar mängden minne som används av modulen. Siffran i kolumnen "Used by" anger hur många andra moduler som använder modulen. Namnen på de andra modulerna som använder modulen kan också listas i "Used by"-kolumnen, men det är ofta inte komplett:



```
sysadmin@localhost:~$ lsmod
Module                Size  Used by
isofs                  40960  0
dccp_diag             16384  0
dccp                   73728  1 dccp_diag
tcp_diag              16384  0
udp_diag              16384  0
inet_diag             20480  3 tcp_diag,dccp_diag,udp_diag
unix_diag             16384  0
xt_nat                16384  1558
xt_tcpudp             16384  1641
veth                  16384  0
vxlan                 49152  0
...
```

Moderkortet i många datorer innehåller det som kallas BIOS, eller Basic Input and Output System. System Management BIOS, eller SMBIOS, är en standard som definierar datastrukturerna och hur information om datorhårdvara kommuniceras.

Kommandoverktyget **fdisk** är användbart för att identifiera och hantera lagringsenheter i ett system. Eftersom det kan användas för att skapa, formatera och ta bort partitioner samt för att hämta information, bör det användas med försiktighet i administratörsläge för att undvika dataförlust. **fdisk** kan användas på två sätt: interaktivt och icke-interaktivt.

När flaggan **-l** används med **fdisk** kommer kommandot att icke-interaktivt lista blockenheter, vilket inkluderar diskar (hårddiskar) och logiska volymer.

Utan flaggan **-l** går **fdisk** in i ett interaktivt läge som vanligtvis används för att ändra partitioner på en diskenhet.

Kör **fdisk**-kommandot för att lista diskenheter. Med flaggan **-l** listas partitions-tabeller för de angivna enheterna och avslutas sedan. Om inga enheter anges används de som nämns i **/proc/partitions** (om den filen finns):

```
fdisk -l
```

```
sysadmin@localhost:~$ fdisk -l
Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040
sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000571a2

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           2048     39845887    19921920   83  Linux
/dev/sda2             39847934    41940991     1046529    5  Extended
/dev/sda5             39847936    41940991     1046528   82  Linux swap
```