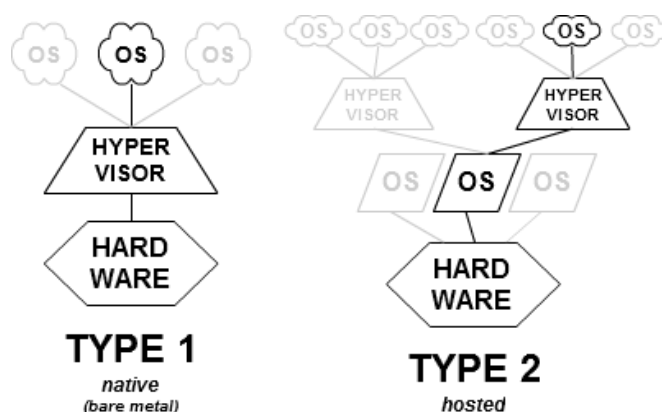


# Labo computernetwerken – Voorbereiding labo's

Labo's "Computernetwerken" veronderstellen van elke student een basiskennis werken met Linux. Deze (zelfstudie) les introduceert Linux basisvaardigheden. We zien de basiswerking van een **virtuele machine** (VM). Eens je een VM met **Linux** op de computer hebt geïnstalleerd, beschik je over een werkomgeving waarin je de basisvaardigheden om met Linux te werken zelf kan aanleren.

## VM-installatie

Een virtuele machine (VM) kan aangemaakt worden in een stuk software, dat een volledige computer emuleert: een hypervisor. Sommige hypervisors zijn geïntegreerd met het operating system (type I, e.g. VMWare ESXi of Microsoft Hyper-V), andere hypervisors zijn applicaties die draaien bovenop gekende operating systems (type II, e.g. VMware Workstation, Linux KVM of VirtualBox).



De software die we zullen gebruiken om VMs te installeren tijdens deze labo's is VirtualBox – een type II hypervisor. Je vindt dit op <https://www.virtualbox.org/wiki/Downloads>.

Eens geïnstalleerd kan je een Linux VM importeren – de URL voor deze VM vind je op Ufora.

## Linux basis

### 1) Basiscommando's Linux

Indien je volledig nieuw bent met het Linux Operating System en de Linux Shell, dan kunnen de volgende commando's nuttig zijn:

**cd <directory>** : verandert de huidige directory naar <directory>. Als je het commando cd uitvoert zonder parameter, dan wordt de home-directory van de gebruiker gekozen.

```
user@linux:~$ cd voorbeeld
user@linux:~/voorbeeld$ cd
user@linux:~$
```

Om naar de home-directory van de huidige gebruiker te verwijzen wordt een tilde (~) gebruikt. Om naar een bestand "file2" in de home directory te verwijzen kunnen we dus "~/file2" gebruiken.

De basis van alle bestandssystemen onder Linux wordt aangeduid met "/", ook wel de filesystem "root" (of wortel) genoemd (niet te verwarren met de gebruiker root). De home directories van de gebruikers bevinden zich in bijna elke Linux installatie in "/home/".

Ook voor de huidige directory en de parent directory zijn korte notaties voorzien: "." voor de huidige directory en ".." voor de parent directory.

```
user@linux:~/voorbeeld$ cd ~
user@linux:~$ cd labo
user@linux:~/labo$ cd ..
user@linux:~$
```

<sup>1</sup> Bron: <https://en.wikipedia.org/wiki/Hypervisor>

**pwd** : geeft de directory weer waarin je aan het werken bent.

```
user@linux:~/voorbeeld$ pwd
/home/user/voorbeeld
user@linux:~/voorbeeld$
```

**mkdir <directory>** : maakt een directory genaamd <directory>

```
user@linux:~$ mkdir voorbeeld
user@linux:~$
```

**ls**: toont een lijst van bestanden van de huidige directory (zoals het *dir* commando in DOS)

```
user@linux:~/voorbeeld$ ls
voorbeeld1 voorbeeld2
user@linux:~/voorbeeld$
```

**echo <string>** : geeft een lijn tekst weer op het scherm. Dit commando kan handig zijn om de waarde van shell variabelen op te vragen:

```
user@linux:~$ echo hello
hello
user@linux:~$ echo $HOME
/home/user
```

**cat <textfile>** : geeft de inhoud van <textfile> weer op het scherm.

```
user@linux:~/voorbeeld$ cat voorbeeld1
Dit is de inhoud van het bestand voorbeeld1
user@linux:~/voorbeeld$
```

**less** : een eenvoudige viewer die toelaat files te bekijken en te doorzoeken. *less* biedt bovendien de mogelijkheid om heen en weer te gaan in de file, iets dat met *cat* niet kan. Als *less* geen argument meekrijgt, wordt de console-input weergegeven. Dit laatste is vooral handig om de output van een commando weer te geven als deze output meer dan 1 scherm lang is (zie ook het doorsturen van output adhv een pipe). Halfweg een document **less** verlaten kan door op **q** te drukken.

**nano <textfile>** : simpele editor in de command line, zodat je veranderingen kan aanbrengen in het bestand <textfile>. Minder krachtig als **vi**<sup>2</sup>, maar eenvoudig in gebruik.

```
^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page
^K Cut Text      ^C Cur Pos       ^X Exit          ^J Justify
^W Where is      ^V Next Page     ^U Uncut Text    ^T To Spell

Ctrl+O bestand opslaan
Ctrl+X editor verlaten
```

**man <commando>** : lezen van de manual pagina's van <commando>

```
user@linux:~/voorbeeld$ man ls
```

Wie ervaart dat zij of hij te weinig achtergrond op de CLI heeft, kan zijn voorkennis bijspijkeren aan de hand van deze website: <http://linuxcommand.org/>. Aanrader voor wie Linux compleet nieuw is! Het luik rond starten met de Shell is relevant; het schrijven van Shell Scripts valt buiten wat verwacht wordt van iemand die start met Linux.

---

<sup>2</sup> **vi** is een krachtige editor die je vind op elk Linux systeem. Het programma werkt echter niet intuïtief, een leercurve moet doorlopen worden om er mee te kunnen werken. **vimtutor** (op linux CLI) neemt ongeveer een uur in beslag. Met **apt install vim** kan je dit op je systeem installeren.

## 2) Handige Shortcuts:

**ctrl+A** en **ctrl+E**: verplaatst de cursor naar het begin of het einde van de command line

```
user@linux:~/voorbeeld$ cat voorbeeld1 [ctrl+A]
user@linux:~/voorbeeld$  at voorbeeld1 [ctrl+E]
user@linux:~/voorbeeld$ cat voorbeeld1 [ctrl+A]
```

**tab** : commando of bestandsnaam vervolledigen. Indien er meerdere mogelijkheden beschikbaar zijn, zal het systeem je verwittigen met een geluid. Indien er veel mogelijkheden beschikbaar zijn, zal het systeem je vragen of alle mogelijkheden getoond mogen worden

```
user@linux:~$ cd voor [tab]
user@linux:~$ cd voorbeeld/
```

**tab tab** : toont bestand(en) of commando('s) die kunnen vervolledigd worden

```
user@linux:~/voorbeeld$ nano voor [tab tab]
voorbeeld1 voorbeeld2
user@linux:~/voorbeeld$ nano voorbeeld
```

## 3) Minder evidente commando's

**grep <pattern>** en **grep <pattern> <file>** : doorzoekt een gegeven tekst op een gegeven patroon.

Elke regel waarin het patroon voorkomt, wordt uitgeschreven. Als er geen file(s) worden opgegeven, dan wordt de console-input doorzocht.

```
user@linux:~$ cat /tmp/test
hello world
yeah yeah yeah
user@linux:~$ grep yeah /tmp/test
yeah yeah yeah
user@linux:~$ grep world /tmp/test
hello world
```

**find <waar zoek je> <wat zoek je>**: doorzoekt een bepaalde map en gaat op zoek naar het patroon van documenten dat je aan het zoeken bent. **Find** laat toe te gaan filteren op wat je normaal te zien zou krijgen met **ls**.

Het voorbeeld hieronder gaat op zoek naar alle directories (**type d**) in de huidige map (**./**), en zoekt slechts één niveau diep (**maxdepth 1**)

```
user@linux:~$ cd /var/log
user@linux:/var/log$ find ./ -type d -maxdepth 1
./
./apt
./dist-upgrade
./fsck
./lxd
./unattended-upgrades
```

#### 4) Input/Output Redirection

In een Linux shell kan de output van een commando makkelijk omgeleid worden naar een bestand. Om tekst in een file weg te schrijven kunnen we gebruik maken van > (overschrijven) of >> (aanvullen):

```
user@linux:~$ echo Hello > /tmp/test
user@linux:~$ cat /tmp/test
Hello
user@linux:~$ echo World >> /tmp/test
user@linux:~$ cat /tmp/test
Hello
World
user@linux:~$
```

We kunnen de inhoud van een tekstbestand ook gebruiken als input voor een commando met < :

```
user@linux:~$ cat /tmp/test
hello world
yeah yeah yeah
user@linux:~$ grep yeah < /tmp/test
yeah yeah yeah
user@linux:~$
```

Het is vaak ook handig om de uitvoer van het ene commando te gebruiken als invoer voor een ander commando. Hiervoor kunnen we een pipe (|) gebruiken:

```
user@linux:~$ cat /tmp/test
hello world
yeah yeah yeah
user@linux:~$ cat /tmp/test | grep yeah
yeah yeah yeah
user@linux:~$
```

Een pipe kan je ook meerdere keren na elkaar gaan gebruiken, tot je de gewenste uitkomst krijgt. Bvb.

```
user@linux:~$ cat /etc/services | grep 80 | head -2
http      80/tcp    www       # WorldWideWeb HTTP
http      80/udp             # HyperText Transfer Protocol
```

#### 5) Meer informatie

Meer nuttige informatie over Linux commando's kan gevonden worden op volgende website:

- [http://www.linuxcommand.org/lc3\\_learning\\_the\\_shell.php](http://www.linuxcommand.org/lc3_learning_the_shell.php)
- <http://www.cheat-sheets.org/saved-copy/The%20One%20Page%20Linux%20Manual.pdf>  
Enigszins verouderd, maar bevat wel nog steeds een mooi overzicht van veel commando's