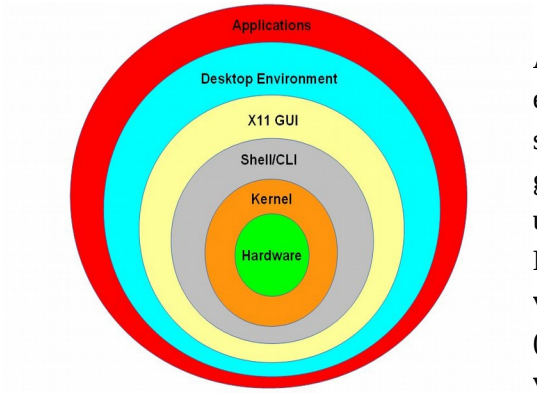
Waarom in een terminal werken?

Een vraag die door menig nieuwe Linux gebruiker gesteld wordt, is waarom er zoveel nadruk gelegd wordt op het gebruik van de terminal om opdrachten uit te voeren.
In een Windows omgeving, waar er voor alles een GUI is, is de ervaring van de modale Windows gebruiker met de CLI zeer beperkt, om niet te zeggen nihil en zijn de voordelen van het gebruik van de terminal dan ook niet gekend.

Het lijdt geen twijfel dat voor het gewone werk (surfen, tekstverwerking, fotobeheer, muziek afspelen, …) de grafische programma's gebruikt worden. Maar voor bepaalde systeem technische zaken en het beheer van je systeem is de terminal in heel veel gevallen de betere partner.

De kip en het ei



Als je de hiërarchie van een systeem zoals Linux bekijkt, dan heb je eerst en vooral de Shell./CLI. Je kunt een volwaardige Linux server/workstation draaien zonder gebruik te maken ook maar 1 grafisch programma, al is het bewerken van foto's logischerwijs wel uitgesloten.
Maar je kunt het Internet (lynx) op, mail sturen en ontvangen (Mutt), volwaardige desktop publishing doen (Latex), programma's draaien (C – Python – Java …), muziek afspelen (CMus) en natuurlijk het volledig beheer van je systeem (desktop zowel als server).
Dat was gewoon de normale manier van werken tijdens de eerste jaren van Linux. Pas later werd hierboven een grafische laag geplaatst via het X Windows Systeem en vandaag heb je dus volwaardige systemen dankzij grafisch ondersteunende systemen zoals Gnome, KDE, Unity, XFCE, LXDE, Mate, Cinnamon, …

Windows heeft in het begin van deze eeuw het roer omgegooid en van een systeem dat eerst DOS laadde en toen met het commando 'win' Windows 95 opstartte, naar een systeem gestapt dat net omgekeerd werkt. Vanaf XP komt de grafische schil eerst en de terminal (dosbox) draait binnen de grafische schil en heeft maar beperkte mogelijkheden.

> *Je kan dus zeggen dat de natuurlijke habitat van een Linux omgeving de terminal is, of in gewone mensentaal: voor alles onder Linux is er een terminal commando, voor bepaalde zaken bestaat er ook een grafische versie ervan.*

Dat kan best, maar in de 21ste eeuw moet het toch anders kunnen

Nog niet overtuigd? Wel laat me proberen het je toch uit te leggen.
Een terminal is niet zomaar een zwart scherm met een blinkende cursor in de linkerbovenhoek.

Er zit een hele hoop intelligentie geprogrammeerd onder die zwarte achtergrond. Hoe krachtiger deze intelligentie, hoe meer je kan doen met de terminal.

Onder GNU besturingssystemen wordt die intelligentie een **shell** genoemd of **Command Line Interpreter (CLI)**. Er bestaan meerdere shells voor Linux, maar de standaard shell is **BASH**. Onder Windows heb je *cmd.exe*.

Bash staat voor **B**ourne **A**gain **SH**ell[1]

Bash is vele malen krachtiger dan cmd.exe onder Windows en dus kan je onder Linux merkelijk meer doen met een terminal dan wat je in Windows onder een zogenaamde dosbox kunt.

Om te weten welke versie van BASH je systeem runt, tik je **bash –version** in je terminal in.

Wat zijn nu de grote voordelen van een BASH-commando vergeleken met een GUI-equivalent?

Want daar gaat het hem in se om. Als ik de moeite wil doen om te leren werken met BASH, moet ik toch iets voor terug krijgen.

Hieronder enkele grote voordelen van BASH:

- In BASH kun je gewone simpele commando's ingeven tot en met volledige scripts. Dat betekent dat een absolute beginner met commando's in BASH kan werken tot en met de gevorderde gebruiker.
  Een script is een bestand dat een opeenvolging van BASH commando's bevat samen met typische programmeer zaken zoals conditionals (if-then-else), pipes, loops, functies, … dat tot een volledig uitvoerbaar programma kan uitgroeien. Je kunt het een beetje vergelijken met de BATCH-scripts onder Windows maar dan vele, vele malen krachtiger.
- BASH is gestandaardiseerd door POSIX en de Single Unix Specification, wat inhoudt dat een script dat je schrijft voor één computer bijna 100% zeker zal werken op andere POSIX machines (zolang je je houdt aan de standaard commando's, en dat zijn er **veel**).
- Voor sommige zaken zijn er gewoonweg nog geen grafische front-ends ontwikkeld en zullen er ook nooit ontwikkeld worden, tenzij je zelf aan de slag wil gaan. In dat geval moet je wel terugvallen op scripts.
- Helpen, tutorials schrijven – zoals ik doe – is veel gemakkelijker als je de juiste commando's kan geven. Niet alleen is het voor de auteur minder werk (bv. geen screenshots, die bij iedere distro anders kunnen zijn), maar iedereen die Linux draait kan gebruik maken van je tutorial.
- Door via de CLI te werken, krijg je meestal een beter inzicht in wat je doet en wat er aan het gebeuren is. Hoe dikwijls gebeurt het dat je niet weet of je programma nu hangt als je grafisch programma geen teken van leven meer toont. In een terminal zie je duidelijker wat er gebeurt.
- Tenslotte is BASH vele malen sneller dan de GUI equivalent. Het is gewoonweg sneller om enkele toetsaanslagen te doen en `TAB` te drukken, dan je een weg te banen door menu's met de muis.

1 The name is an acronym for the 'Bourne-Again SHell', a pun on Stephen Bourne, the author of the direct *ancestor* of the current Unix shell sh, which appeared in the Seventh Edition Bell Labs Research version of Unix.

Enkele voorbeelden van BASH

Veel zaken die je via een terminal doet kun je niet doen via een GUI.

Hernoemen 'en masse'

`rename` is een leuk programma dat je toelaat tientallen bestanden te hernoemen gebaseerd op een patroon. Een simpel voorbeeld ervan: `rename s/myfile([0-9]+).txt /$1-myfile.txt/` vervangt alle bestanden met een bestandsnaam als 'bond007.txt' in '007-bond.txt', ongeacht het nummer. Wie weet zijn er bij jou MP3's die je moet hernoemen.

Werken op bestanden

Hoe dikwijls gebeurt het niet dat we willen weten hoe groot een bepaald bestand is, voordat we het willen kopiëren? Je kunt natuurlijk Dolphin of Nautilus openen, klikken naar de map van het betreffende bestand, en dan met een rechter muisklik de eigenschappen bekijken. Of…
Als extraatje maak ik gebruik van het commando `time` dat je aangeeft hoe lang een bepaald commando nodig heeft.

- Hoeveel verbruikt je complete home folder?

```
$ time du -hs ~
179G    /home/alain

real    0m0.234s
user    0m0.092s
sys     0m0.142s
```

- Hoeveel ruimte is er nog over op mijn harde schijf? Dit commando is een leukerd omdat het `egrep` gebruikt om de tijdelijke mappen (tmp-folder bv.) uit te sluiten plus het laat je ook de bestandstypes zien:

```
$ time df -hT | egrep -i "file|^/"
/dev/sda8       ext4         19G     5,8G   12G  34% /
/dev/sda6       ext4        282G     182G   86G  68% /home

real    0m0.004s
user    0m0.003s
sys     0m0.003s
```

- Welke bestanden zijn er op een bepaalde datum veranderd?

```
alain@desktop ~/Downloads $ time !!
time ls -lrt | awk '{print $6" "$7" "$9 }' | grep 'jul 18'
jul 18 w_makb09.pdf
jul 18 w_mach02.pdf
jul 18 w_advb01(1).pdf
jul 18 w_dive01.pdf
jul 18 linuxmint-17-cinnamon-64bit-v2.iso.torrent
jul 18 linuxmint-17-cinnamon-64bit-v2.iso
jul 18 456
jul 18 linux-brprinter-installer-2.0.0-1
jul 18 dcp7065dnlpr-2.1.0-1a.i386.deb
```

```
jul 18 cupswrapperDCP7065DN-2.0.4-2a.i386.deb
jul 18 uninstaller_DCP7065DN
jul 18 uninstaller_brscan4
jul 18 uninstaller_brscan-skey

real 0m0.008s
user 0m0.007s
sys 0m0.008s
```

- Wat zijn de grootste bestanden op mijn systeem? Via sudo su naar de root prompt #

```
desktop alain # time find / -type f -print0| xargs -0 ls -s | sort -rn | awk
'{size=$1/1024; printf("%dMb %s\n", size,$2);}' | head -5

1730Mb /home/alain/Video's/Relaxing/Relaxation
1426Mb /home/alain/Video's/Star
1393Mb /home/alain/Video's/Relaxing/Relaxing
1383Mb /home/alain/dwhelper/Keane
1322Mb /home/alain/Downloads/iso/linuxmint-17-xfce-dvd-64bit.iso

real 0m10.034s
user 0m9.856s
sys 0m2.263s
```

Pas goed op.

De terminal kan en is een gevaarlijk mijnenveld.

- **Als root ben je GOD. God mag alles en jij als root op je systeem dus ook. Denk geen 2x na, maar 5x na als je als root een commando ingeeft.**
- **Doe altijd eerst het nodig opzoekwerk.**
- **Bedenkt heel goed wat je gaat doen voordat grote aanpassingen te doen, en overloop goed alle opties, tekentjes, spaties, … die je hebt ingetikt.**
- **Denk niet dat je plots alles kan. Je leert iedere dag iets onder Linux.**
- **Back-up, back-up, back-up, maak regelmatig een backup van je /home folder want dat spaart uren werk uit als er iets misloopt.**

Om af te sluiten.

Het werken in een terminal lijkt vooral in het begin vreemd, onhandig, moeilijk en overbodig. Alleen door er regelmatig mee te experimenteren en je gevonden commando's via een alias in te korten, leer je de ware kracht van BATCH kennen.

In een latere fase kun je dan je commando's groeperen in kleine bestandjes – scripts – en dan gaat nog een mooiere wereld open.