

Command line fun

Door: Merijn Vogel

Datum: 2023-03-09

Inleiding

Je linux geïnstalleerd of je hebt een server verkregen. Je kijkt tutorials en wat blijkt: veel linuxgebruikers houden van de `command line` of in het Nederlands: `opdrachtregel`.

Wat is de command line?

De command line is een schil (`shell`) vanuit waar je de computer opdrachten kunt geven.

In feite is het een soort programmeertaal die als doel heeft programma's te starten en beheren.

Waarom een command line?

In het tijdperk voor windows (de pre-historie, voor ca 1990) waren er weinig grafische omgevingen waar je met een aanwijzer (muis) programma's kon starten. Een muis was een luxe of simpelweg nog niet uitgevonden. Een grafische omgeving zoals windows bestond op slechts enkele systemen. En in de echte prehistorie, zoals de jaren zeventig, was een toetsenbord en scherm (of zelfs `teletype`) de enige manier om iets van een computer gedaan te krijgen.

Command lines in windows

Windows kent vele command lines. De "klassieke" is `cmd`. Dat geeft je een prompt die erg op `DOS` lijkt. Dan is er `powershell`, een krachtige schil met opdracht regel die zo'n beetje heel `.Net` als programmeertaal heeft. Ook kun je zelf bijvoorbeeld `git-bash` installeren, dat is een linux shell maar dan in windows!

Wat kan je met de command line?

Programma's starten, scripts schrijven en nog veel meer leuks.

Hoe doe ik...

Typische commando's

```
[meriadoc@merijn-lenovo Quadraam]$ pwd
/home/meriadoc/MerijnCloudBigStorage/Quadraam
[meriadoc@merijn-lenovo Quadraam]$ ls
2021-2022  2022-2023  2022-blok-4
[meriadoc@merijn-lenovo Quadraam]$ cd 2022-2023
[meriadoc@merijn-lenovo 2022-2023]$ ls
databases  linux  mega-query.txt
[meriadoc@merijn-lenovo 2022-2023]$ cd linux
[meriadoc@merijn-lenovo linux]$ ls
'Q-highschool informatica_ linux en servers - les
1-20230126_144019-Meeting Recording.mp4'
'Q-highschool informatica_ linux en servers - les
```

```
1-20230126_153104-Meeting Recording.mp4'
```

```
'Wat kun je over linux en servers leren.pdf'
```

- `pwd`: print working directory, oftewel, waar in het file-system ben ik.
- `ls`: list files (en directories), "kijk om je heen"
- `cd`: change directory, navigeren

Een programma starten

Als firefox als browser is geïnstalleerd, kun je dat starten op de command line:

```
$ firefox
```

Nu wordt firefox gestart en in het venster met de command line verschijnt soms allerlei ellende:

```
[meriadoc@merijn-lenovo ~]$ firefox
ATTENTION: default value of option mesa_glthread
overridden by environment.
ATTENTION: default value of option mesa_glthread
overridden by environment.
ATTENTION: default value of option mesa_glthread
overridden by environment.
ATTENTION: default value of option mesa_glthread
overridden by environment.
[2023-03-09T11:48:40Z ERROR mp4parse] Found 2 nul
bytes in "\0\0"
[2023-03-09T11:48:40Z ERROR mp4parse] Found 2 nul
bytes in "\0\0"
```

```
[2023-03-09T11:48:40Z ERROR mp4parse] Found 2 nul
bytes in "\0\0"
[2023-03-09T11:48:40Z ERROR mp4parse] Found 2 nul
bytes in "\0\0"
[2023-03-09T11:49:28Z ERROR
viaduct::backend::ffi] Missing HTTP status
[2023-03-09T11:49:28Z ERROR
viaduct::backend::ffi] Missing HTTP status
```

Programmeurs kunnen hier misschien ooit wat mee, maar eigenlijk is dit vooral "lawaai".

Ook komen we niet terug bij die `$`, de prompt van de computer waar die ontvankelijk is voor jouw invoer. Dat gebeurde hierboven bij `ls` en `pwd` wel. Dit komt doordat firefox niet meteen klaar is, niet stopt. Om toch firefox te kunnen starten vanaf de command line en er daarna nog gebruik van te maken, kun je het starten in de achtergrond.

Een programma starten in de achtergrond

Het magische karakter hiervoor is de ampersant: `&`

```
[meriadoc@merijn-lenovo ~]$ firefox &
[1] 19676
[meriadoc@merijn-lenovo ~]$ ATTENTION: default
value of option mesa_glthread overridden by
environment.
ATTENTION: default value of option mesa_glthread
overridden by environment.
ATTENTION: default value of option mesa_glthread
```

overridden by environment.

ATTENTION: default value of option mesa_glthread overridden by environment.

```
[2023-03-09T11:58:45Z ERROR
```

```
glean_core::metrics::ping] Invalid reason code startup for ping background-update
```

(Wat een ellende weer... firefox kan ook nooit eens stil zijn).

Poging 2 maar dan iets anders starten:

```
[meriadoc@merijn-lenovo ~]$ gnome-mines &
```

```
[1] 2305
```

```
[meriadoc@merijn-lenovo ~]$
```

Da's beter. Gnome-mines (een minesweeper variant) is gestart. Het getal tussen blokhaken is een nummer van de job die je gestart hebt. Het getal 2305 is het proces-id.

lets stoppen vanaf de command line

Stel, je start `gnome-mines` maar niet in de achtergrond. Zit je dan met je terminal.

Om het in de achtergrond draaiende programma te stoppen kun je `ctrl-c` typen.

```
[meriadoc@merijn-lenovo ~]$ gnome-mines
```

```
^C
```

```
[1]+  Done                               gnome-mines
```

```
[meriadoc@merijn-lenovo ~]$
```

Bladeren door geschiedenis

Na een tijdje heb je best wat commandos uitgevoerd. Met de cursor up pijl kun je het vorige commando oproepen, en door dat te herhalen alle vorige commandos.

Wil je een groter overzicht hebben van je hele geschiedenis van commandos, dan is daar (hoe kan het anders) een commando voor: `history`:

```
[meriadoc@merijn-lenovo ~]$ history | tail -n 10
495  clear
496  yay -S minesweeper-tui
497  yay -S mines
498  yay -S minesweeper-tauri-git
499  yay -Scc
500  syncthing&exit
501  gnome-mines &
502  gnome-mines
503  history
504  history | tail -n 10
[meriadoc@merijn-lenovo ~]$
```

Alle 500 vorige commando's, deels in eerdere sessies, zijn beschikbaar. Hier zijn weer trukerijen mee mogelijk waar zelfs mijn directe collega's hoofdschuddend naar kijken...

(PS: de `|` wordt in de volgende sectie behandeld over uitvoer doorsturen)

Uitvoer doorsturen

Een belangrijk ding van linux, en daarin is het beter dan de command lines in windows, is het feit dat je uitvoer van

programma 1 kunt doorsturen naar de invoer van programma 2.

Hierboven deed ik dat met `history | tail -n 10`.

`tail` is een utility die invoer opeet en alleen de laatste `n` regels op de uitvoer afdrukt. `history` gaf meer dan 500 regels, dus dat was me wat veel.

`grep` is een utility die kan zoeken naar tekst. Dus als ik alle vorige aanroepen naar firefox in mijn geschiedenis wil terugvinden:

```
[meriadoc@merijn-lenovo ~]$ history|grep fire
480  firefox
494  firefox &
505  history|grep fire
[meriadoc@merijn-lenovo ~]$
```

In het dagelijks leven als linux beheerder maak ik soms verrassend lange ketens van commando's om te krijgen wat ik wil.

Conclusie

Als je weet hoe het heet, kun je het starten vanaf de command line

Dat is wel een nadeel. In plaats van "die webbrowser met dat oranje vosje", moet je weten dat die `firefox` heet. En dat gaat nog, maar de minesweeper variant op mijn laptop heet `gnome-mines`. Dat ben ik gauw genoeg vergeten... (Gnome

komt van een grafische omgeving, de gnome desktop, met zijn eigen verzameling "handige" bibliotheken om grafische programma's mee te schrijven, die library heet `QT` .)

| Uitvoer doorsturen is nuttig

De gedachte achter command line linux tooltjes is: doe 1 ding en doe dat goed. Dus zijn de meeste programma's geen enorm Zwitsers zakmes, maar eerder een enkel goed koksmes. Door een keten van programma's te maken kun je het resultaat krijgen dat je nodig hebt.

| Er is meer!

Door een bestand te maken met commando's en dat uitvoerbaar te maken (zie mijn verhaal over [Voorbeelduitwerking-bestanden-en-permissies](#)), kun je meerdere commando's achter elkaar gebruiken. De shell kent ook variabelen en allerlei ander moois.

De megagrote handleiding van de standaard shell `bash` is nogal lang:

```
[meriadoc@merijn-lenovo ~]$ man bash | wc
6399    50285   382998
```

Ja, er is een tooltje dat `wc` heet. Het staat voor wordcount. Dit telt regels (6399), woorden (50285) en letters. Dit verhaal is iets meer dan 1000 woorden, dus die handleiding is serieus dik.

In mijn studententijd heb ik die uitgebreid bestudeerd... pluk ik nu nog de vruchten van ;-)