

Linux CLI

Philippe Dellaert

Student IT vzw

13 oktober 2009

Overzicht

- 1 Deze Workshop
- 2 Basis
- 3 Bestanden
- 4 Shell scripting

Linux CLI

- 1 Deze Workshop
- 2 Basis
- 3 Bestanden
- 4 Shell scripting

Deze Workshop

Shell commando's worden zo voorgesteld:

Structuur commando

```
commando -opties argumenten
```

Meer info over een commando

```
man <commando>  
<commando> --help  
<commando> -h
```

Regelmatig:

Doe het zelf!

```
commando -opties argumenten  
commando2 -opties argumenten
```

Dit zijn een reeks van commando's die je zelf eens kan proberen in je console.

Doe het zelf!

Doe het zelf!

```
man man
```

Je krijgt de manual te zien van het 'man' commando. Enkele handige toetsen:

Handige key shortcuts

pijltjes down/up → regel naar beneden/boven

page down/up → pagina naar beneden/boven

/<term><enter> → Zoeken op 'term'

q → sluiten

Linux CLI

- 1 Deze Workshop
- 2 Basis**
- 3 Bestanden
- 4 Shell scripting

Basis commando's

Huidige map

```
pwd
```

Inhoud van een map

```
ls <mapnaam>
```

Van map veranderen

```
cd <mapnaam>
```

Map aanmaken

```
mkdir <mapnaam>
```

Leeg bestand aanmaken

```
touch <bestandsnaam>
```

Doe het zelf!

Doe het zelf!

```
pwd
mkdir testmap
ls
cd testmap
pwd
touch testfile
ls -a
```

Doe het zelf!

Resultaat

```
$ pwd
/home/philippe
$ mkdir testmap
$ ls
testmap
$ cd testmap
$ pwd
/home/philippe/testmap
$ touch testfile
$ ls -a
.. testfile
```

Basis commando's deel 2

Bestand copieren

```
cp <bestandnaam> <doelnaam>
```

Bestand hernoemen

```
mv <bestandnaam> <doelnaam>
```

Bestand verplaatsen

```
mv <bestandnaam> <mapnaam>/
```

Bestand verwijderen

```
rm <bestandsnaam>
```

Lege map verwijderen

```
rmdir <mapnaam>
```

Doe het zelf!

Doe het zelf!

```
cp testfile testfile2
ls
mv testfile2 testfile3
ls
mv testfile3 ../
ls ..
cd ..
rm testfile3
rmdir testmap
rm -r testmap
```

Doe het zelf!

Resultaat

```
$ cp testfile testfile2
```

```
$ ls
```

```
testfile testfile2
```

```
$ mv testfile2 testfile3
```

```
$ ls
```

```
testfile testfile3
```

```
$ mv testfile3 ../
```

```
$ ls ..
```

```
testfile3 testmap
```

```
$ cd ..
```

```
rm testfile3
```

```
rmdir testmap
```

```
rmdir: failed to remove 'testmap/': Directory not empty
```

```
rm -r testmap
```

Linux CLI

- 1 Deze Workshop
- 2 Basis
- 3 Bestanden**
- 4 Shell scripting

Bestandinfo, inhoud en zoeken

Bestand info en inhoud

```
file <bestandsnaam>  
less <bestandsnaam>
```

Zoeken naar een bestand

```
find <opties> <basismap> <zoekopdrachten>
```

Doe het zelf!

Doe het zelf!

```
file /var/log
file /var/log/syslog.1.gz
file /var/log/syslog
less /var/log/syslog
```

less heeft dezelfde shortcuts als man

Doe het zelf!

```
find /var/log/ -name 'syslog'
find /var/log/ -iname '*syslog*'
find /var/log/ -type f
```

Doe het zelf!

Resultaat

```
$ file /var/log
/var/log/: directory
$ file /var/log/syslog.1.gz
/var/log/syslog.1.gz: gzip compressed data, was "syslog.0",
from Unix, last modified: Mon Oct 12 07:45:02 2009, max
compression
$ file /var/log/syslog
/var/log/syslog: ASCII text
$ less /var/log/syslog
$ find /var/log/ -name 'syslog'
/var/log/syslog
$ find /var/log/ -iname '*syslog*'
/var/log/syslog /var/log/syslog.1.gz ...
$ find /var/log/ -type f
```

Editors

Bestand editors

vim

nano

Vim key shortcuts

i → Nieuwe tekst toevoegen op huidige locatie

a → Nieuwe tekst toevoegen op locatie achter huidige locatie

<ESC> → Ga uit 'insert' modus (stop met aanpassen)

:q → Vim sluiten

:w → Bestand opslaan

:w naam → Bestand opslaan als 'naam'

:x → Bestand oplaan en vim sluiten

Meer info op: <http://www.fprintf.net/vimCheatSheet.html>

Nano heeft in zijn interface de keys staan.

Doe het zelf!

Doe het zelf!

```
cp /var/log/syslog ./  
vim syslog
```

Bewerk het bestand met de key shortcuts van de vorige slide, test wat, eventueel met de cheatsheet erbij!
In `vim` kan je op dezelfde manier zoeken als in `less` of `man`.
Probeer het eens!

Doe het zelf!

```
nano syslog
```

Bewerk het bestand ook eens hiermee!

Doe het zelf!

```
rm syslog
```

Verborgen bestanden

- Bestanden beginnende met '.' worden standaard niet getoond
- Deze bevatten voornamelijk instellingen van programma's

Doe het zelf!

Doe het zelf!

```
cd ls -a
```

Dit geeft alles weer in je home map, ook de verborgen bestanden.

Rechten op bestanden

- Bestanden hebben altijd een eigenaar en een groep ingesteld
- Er zijn 3 niveau's van rechten:
 - User (eigenaar)
 - Group
 - Others
- Er zijn 3 type rechten:
 - `r` → Read
 - `w` → Write
 - `x` → Executable
- Een map moet altijd 'Executable' zijn voor het juiste niveau als dat niveau moet kunnen zien wat erin zit

Rechten op bestanden

Voorbeelden

`chmod u+rw <bestandsnaam>` → Geef read/write rechten aan User

`chmod go+r <bestandsnaam>` → Geef read rechten aan Group en Others

`chmod go-wx <bestandsnaam>` → Neem write/executable rechten af van Group en Others

`chmod o-rwx <bestandsnaam>` → Neem alle rechten af van Others

`chmod a-rwx <bestandsnaam>` → Neem alle rechten af van iedereen

Linux CLI

- 1 Deze Workshop
- 2 Basis
- 3 Bestanden
- 4 Shell scripting**

Inleiding

- Bash scripting
- Enkel basis
- Veel complexe mogelijkheden
- Goede guide: <http://tldp.org/LDP/abs/html/>

Voorbeelden

Simpele loop op de bestanden in de map

```
for i in *; do
  echo $i;
done;
```

Simpele conditional statement

```
a = 1
if [ $a -gt 10 ]; then
  echo "A is bigger than 10";
else
  echo "A is smaller than 10";
fi;
```

Output van een commando in een variabele steken

```
varname = `commando -opties argumenten`
```

Groot voorbeeld

Alles samen + een argument

```
for i in `seq 1 $1`; do
  if [ $i -lt 10 ]; then
    k="0$i";
  else
    k=$i;
  fi;
  mv archive.part$k.rar archive.r$k;
done;
```