

Besturingssystemen: Linux

HoGent Bedrijf en Organisatie

2020-2021

Intro

Academiejaar 2020-2021

Lectoren:

- Nathalie Declercq
- Thomas Pollet
- Andy Van Maele
- Bert Van Vreckem

Waarom Linux?

Ik ga sowieso geen systeem- en netwerkbeheer volgen, waarom moet ik me dan hier mee bezig houden?

Linux is overal

... en je kan er gewoon niet meer omheen

- Grootste websites
 - webdev => jouw apps zullen op Linux draaien
 - e-business: Drupal
- MacOS X
 - Ook UNIX
 - Bash-shell + commando's

-
- Moderne tools voor software release management
 - Continuous Integration/Delivery
 - Docker
 - Embedded systemen
 - Arduino, Raspberry Pi, ...
 - Multimedia-apparatuur, TV's, NAS, ...
-

Ik wil me specialiseren in Windows. Linux interesseert mij niet

Ook in de Windows-wereld

- Azure cloud platform draait >50% Linux VMs
 - Microsoft loves Linux!
-

The '70's called, they want their terminal back!

De kracht van de CLI

- Command line interface (CLI) is de sleutel tot **automatisering**
 - Miljoenen servers beheren gaat niet via GUI
 - Ook in Windows ziet men dit nu in (PowerShell)
- CLI Instructies zijn **bondiger** en makkelijker reproduceerbaar dan GUI instructies

Leerpad Linux

- Overzicht hoofdstukken
- Per hoofdstuk:
 - Voorbereiding
 - Achtergrondinformatie
 - Slides
 - Oefeningen met oplossingen
 - Toetsingsvragen
 - Labo-opdrachten

Verloop contactmoment

- Klassikale instructie
- Werken aan labo-oefeningen
 - individuele feedback
 - extra klassikale uitleg
- Korte intro volgende sessie

Buiten contactmomenten

- Bereid volgende sessie voor
 - Leerpad: voorbereiding, achtergrondinfo, toetsingsvragen
 - Werk verder aan labo's
-

Van een sysadmin wordt **zelfstandigheid** verwacht

De werkwijze in deze cursus is een eerste stap in die richting!

Hoofdstuk 1. Linux installeren

Laboverslagen bijhouden

- Gebeurt ahv Github repo met startinfo
- Ga naar Chamilo, klik door op link naar *classroom.github.com/..*
- Je krijgt persoonlijke, private Github-repo voor laboverslagen, scripts, ...

Software downloaden

- Via de websites
 - Traag, WiFi overbelast
- <http://tin.hogent.be/public/>
- USB-stick doorgeven

Linux installeren

1. Installeer VirtualBox, Git client
 - Leerpads Besturingssystemen, 6.1 "Installatie Software"
 - Leerpads Linux, 1.2 "Achtergrondinformatie"
2. Maak nieuwe VM voor Linux
 - Labo 1 - via leerpads (1.5) of Github repo
 - Volg de instructies!
3. Installeer Fedora

Hoofdstuk 2. Linux leren kennen

Hulp zoeken (1)

```
# Hulp over het commando 'passwd'
man passwd
```

```
# Hulp over het configuratiebestand /etc/passwd
man 5 passwd
```

```
# Zoek in alle man-pages naar de string 'passwd'
man -k passwd
apropos passwd
```

Hulp zoeken (2)

- Binnen man-page:
 - q - man-page verlaten
 - / - zoeken binnen de pagina
 - n - ga naar volgende zoekresultaat
 - N - ga naar vorige zoekresultaat
- Secties, vb:
 - 1 - commando's
 - 5 - configuratiebestanden
 - 8 - systeembeheercommando's
 - Notatie: vb. passwd(1), passwd(5)

Belangrijke man-pages

... die niet over een commando gaan:

```
# directorystructuur Linux (Filesystem Hierarchy)
man hier
```

```
# "ingebouwde" Bash commando's
```

man builtins

"wildcards" in bestandsnamen (bv. `*.*`, `[a-z]`)

man 7 glob

Structuur van een commandoregel

\$ COMMANDO [OPTIES] ... [ARGUMENTEN] ...

- “Onderdelen” gescheiden door **spaties**
- 1e woord = **commando**
- **Opties** veranderen het gedrag van een commando
 - beginnen met streepje (`-` of `--`)
- **Argumenten** zijn de entiteiten waarop het commando uitgevoerd wordt

Commando

Het eerste “woord” van een opdrachtregel moet een **commando** zijn

- Alias of functie, gedefinieerd door gebruiker
- Ingebouwd in Bash (zie `man builtins`)
- Uitvoerbaar bestand in één van de directories in `${PATH}`
- Absoluut pad naar uitvoerbaar bestand
- `which COMMANDO`

Let op! Bash zoekt nooit in de huidige directory!

Opties

Wijzigen het gedrag van het commando

- Korte notatie: `-a -b -c`
 - Korter schrijven als `-abc`
- Lange notatie: `--foo --bar`

Let op! Niet alle commando's volgen de conventie! (bv `find`)

Substitutie/expansie (1)

Vóór uitvoeren van een commando vervangt Bash bepaalde uitdrukkingen:

- *Brace expansion*, vb. `{1..10}`, `dir/{subdir1,subdir2,subdir3}`
 - vb. `mkdir -p project/{src,lib,build}`
- *Tilde expansion*: `~` wordt vervangen door home-directory, vb. `/home/student/`
 - vb. `ls ~/.ssh`
- *Parameter expansion*: variabelennamen worden vervangen door waarde, vb. `${USER}` -> `student`

Substitutie/expansie (2)

- *Command substitution*: `$(command)` wordt vervangen door uitvoer van commando
 - vb. `date=$(date)`
- *Filename expansion* of “globbing”: wildcards in bestandsnamen, vb. `*`, `?`, `[abc]`, enz.
 - vb. `rm *.class`
- ...

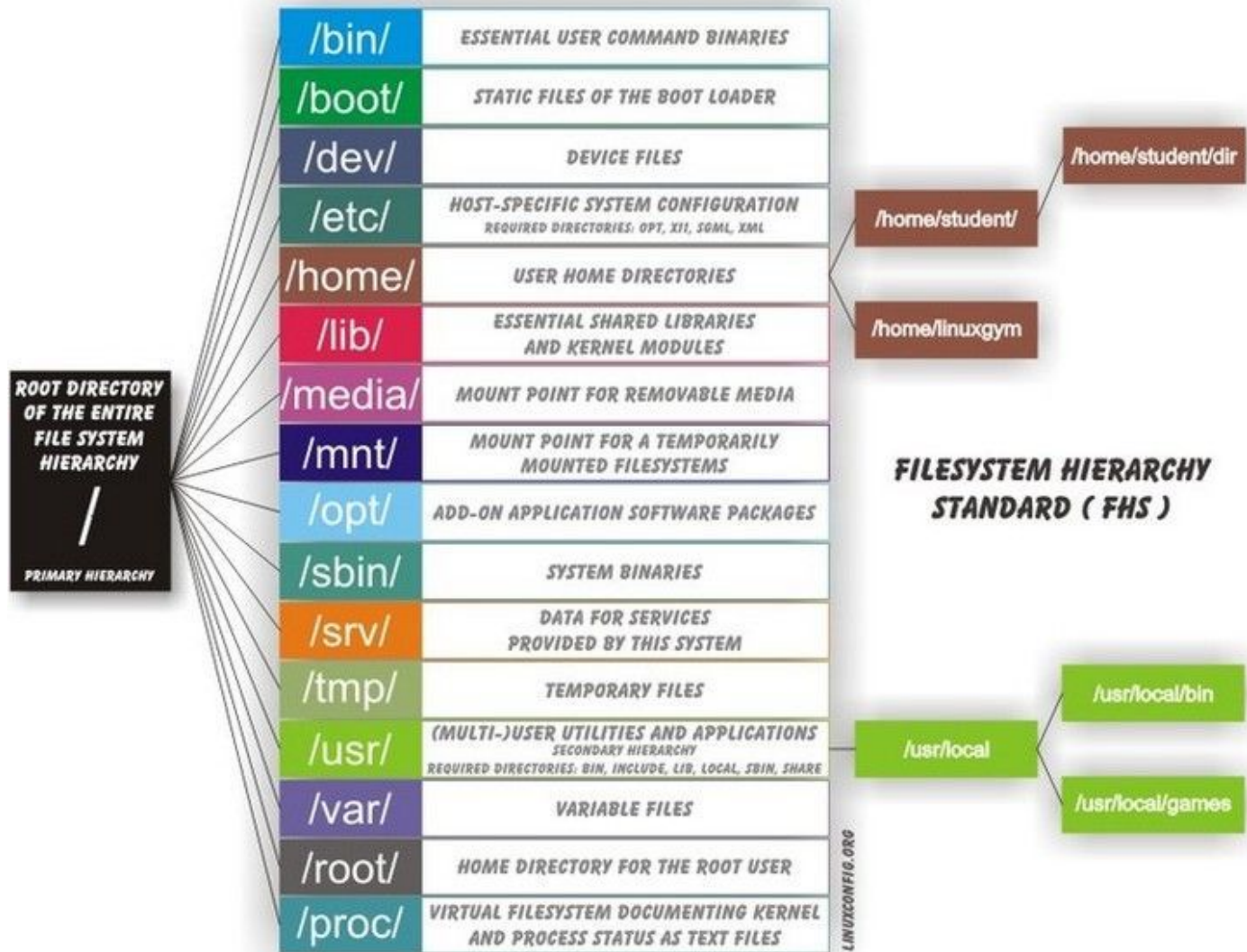
Resultaat expansie

- `set -x` toont resultaat van expansie ("debug mode")
- `set +x` zet optie terug uit

Voorbeeld:

```
set -x; ls -ld ${HOME}/D[eo]*; set +x
set -x; mkdir -p a/{b,c,d}/{e,f}; set +x
```

Linux directorystructuur



Absoluut/relatief pad

- **Absoluut** pad:
 - begint met /
 - ten opzichte van 'root directory' /
 - vb. /home/student, /tmp, /var/www
- **Relatief** pad:
 - begint **niet** met /

- ten opzichte van **huidige** directory (pwd)
- ls Documents, mkdir linux

Speciale directorynamen

/	De "root directory"
.	De huidige directory (pwd)
..	De bovenliggende directory
~	Je "home-directory" (vb. /home/student)

Opm. De "home directory" van de gebruiker "root" is /root

Werken met directories

Commando	Taak
pwd	Toon huidige directory
ls	Toon inhoud huidige directory
cd	Ga naar een andere directory
mkdir	Maak een subdirectory aan
rmdir	Verwijder een lege directory

Werken met bestanden

Commando	Taak
cat	Toon inhoud van een bestand
less	Toon inhoud, per pagina (navigeer met pijltjes)
touch	Maak leeg bestand aan (eigenlijk: pas datum laatste wijziging aan)
cp	Kopieer bestanden
mv	Verplaats bestanden (of hernoemen!)
rm	Verwijder bestanden of directories

Meerdere bestanden opgeven: globbing

Commando uitvoeren op meer dan 1 bestand?

- Geef elk bestand apart op (spatie ertussen)
 - cp a.txt b.doc c.jpg /tmp
- Gebruik *globbing*-patronen
 - cp /media/usbstick/*.jpg ~/Pictures/

Globbering-patronen

Patroon	Betekenis	Voorbeeld
?	Eén willekeurig teken	ls /bin/??
*	Willekeurige string (ook leeg)	ls *.txt, ls a*
[...]	Elk teken opgesomd tussen []	ls /bin/[A+_*]

Patroon	Betekenis	Voorbeeld
[A-Z]	Van A t/m Z	ls /bin/*[A-D1-3]
[! ...]	Niet 1 v/d opgesomde tekens	ls /bin/[!a-z]*

Let op

Globbering ≠ Reguliere expressies!!!

- Globbing:
 - Bestandsnamen opgeven
 - case statement in Bash scripting
 - Véél beperkter dan regex!
- Reguliere expressies:
 - Tekstpatronen *binnen* bestanden opzoeken

Enkele “Anti-patterns”

Gebruik nooit regex om bestanden te selecteren!

```
# Fout:
ls /bin | grep 'a.*'
# Beter:
ls /bin/a*
```

In dit soort gevallen is find overbodig:

```
find . -maxdepth 1 -type f -name 'a*' -exec cp {} /tmp \;
# Beter:
cp a* /tmp
```

Hoe maak je een bestand aan?

1. Met teksteditor Vi/Vim: vim bestand.txt
2. Met teksteditor Nano: nano bestand.txt
3. Leeg bestand: touch bestand.txt

Vim survival guide

- Bij opstarten van Vim kom je terecht in *normal mode*.
- Als je tekst wil invoeren moet je naar *insert mode*.

Taak	Commando
Normal mode -> insert mode	i
Insert mode -> normal mode	<Esc>
Opslaan	:w
Opslaan en afsluiten	:wq
Afsluiten zonder opslaan	:q!

Nano

- Ook een command-line teksteditor
- Iets toegankelijker dan Vim

- Shortcuts staan onderaan het scherm
 - vb. Exit: ^X -> Ctrl+X

Archief aanmaken

vgl. zip-bestand

```
tar cjf documents.tar.bz2 Documents/
tar xf documents.tar.bz2
```

- tar = Tape ARchive
- Opties:
 - c - Create
 - x - eXtract
 - j - compressie met bzip2
 - z - compressie met gZip
 - f - naam van het archiefbestand (File)
 - v - toon uitgepakte bestanden (Verbose)

Hoofdstuk 3. Werken met tekst

Tekst afdrukken met echo

```
echo "Hallo wereld!"
echo "Hallo ${USER}!" # variabelen gebruiken
echo 'Hallo ${USER}!' # enkele aanhalingstekens!
```

Tekst afdrukken met printf

```
printf 'Hallo wereld!\n'
printf 'Hallo %s!\n' "${USER}"
printf 'Het gebruikers-ID van %s is %d\n' "${USER}" "${UID}"
```

- vgl. printf() methode in Java!
- aangewezen in scripts, beter gedefinieerd dan echo

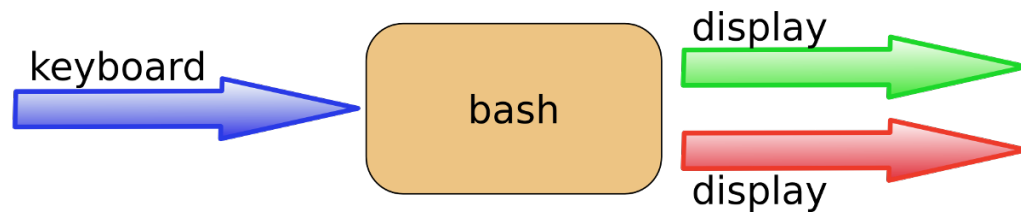
Input en output



- *stdin*, standard input
 - vgl. Java System.in
- *stdout*, standard output
 - vgl. System.out
- *stderr*, standard error
 - vgl. System.err

Normaal gedrag

- *standard input*: invoer toetsenbord
- *standard output/error*: afdrukken op scherm (console)

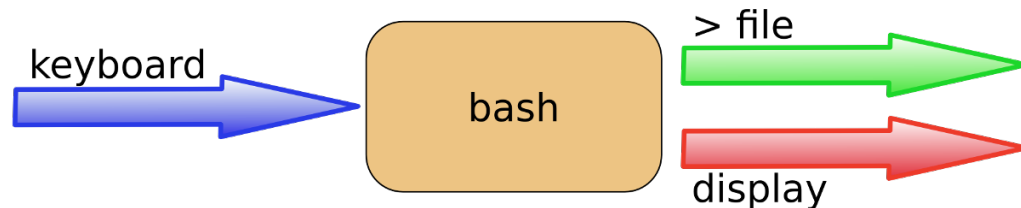


I/O Redirection (1)

Syntax	Betekenis
cmd > file	schrijf uitvoer van cmd weg naar file
cmd >> file	voeg toe aan einde van file
cmd 2> file	schrijf foutboodschappen van cmd weg naar file
cmd < file	gebruik inhoud van file als invoer voor cmd
cmd1 cmd2	gebruik uitvoer van cmd1 als invoer voor cmd2

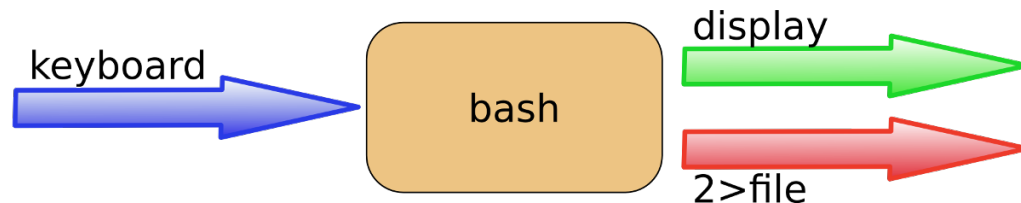
I/O Redirection (2)

cmd > file



I/O Redirection (3)

cmd 2> file



Pipes

Probeer het volgende eens!

```
$ sudo dnf install fortune cowsay lolcat figlet
$ echo ${USER} | figlet
$ fortune
$ fortune | cowsay
$ fortune | cowsay | lolcat
```

Combineren

```
# stdout en stderr apart wegschrijven
find / -type d > directories.txt 2> errors.txt

# stderr "negeren"
find / -type d > directories.txt 2> /dev/null

# stdout en stderr samen wegschrijven
find / -type d > all.txt 2>&1

# invoer én uitvoer omleiden
sort < unsorted.txt > sorted.txt 2> errors.txt
```

Foutboodschappen afdrukken

(Equivalent van `System.err.printf()`)

```
printf 'Error: %s is not a directory\n' "${dir}" >&2
```

Here documents (1)

Als je meer dan één lijn wil afdrukken:

```
cat << _EOF_
Usage: ${0} [OPT] ... [ARG] ..
```

OPTIONS:

 -h, --help Print this help message

```
_EOF_
```

Let op: geen spaties toegelaten vóór de eindemarkering

Here documents (2)

Dit kan bv. ook:

```
mysql -uroot -p"${db_password}" mysql << _EOF_
DROP DATABASE IF EXISTS drupal;
CREATE DATABASE drupal;
GRANT ALL PRIVILEGES ON drupal TO ${drupal_usr}@localhost
IDENTIFIED BY ${drupal_password};
_EOF_
```

Filters

- Filter = commando dat:
 1. leest van stdin of bestand,
 2. tekst transformeert, en
 3. wegschrijft naar stdout
- Combineer filters via | (pipe) om complexe bewerkingen op tekst toe te passen
 - De *UNIX-filosofie*

Filters: overzicht (1)

Commando	Doel
awk	Veelzijdige tool voor bewerken van tekst
cat	Druk inhoud bestand(en) af op stdout
cut	Selecteer "kolommen" uit tekstbestanden
fmt	Herformatteer tekst (bv. bepaald aantal kolommen)
grep	Zoek ahv reguliere expressies naar tekstpatronen in bestanden
head	Toon de eerste regels van een tekstbestand
join	Voeg twee tekstbestanden samen ahv een gemeenschappelijke kolom
nl	Voeg regelnummers toe aan een bestand

Filters: overzicht (2)

Commando	Doel
paste	Voeg twee tekstbestanden regel per regel samen
sed	Veelzijdige tool voor bewerken van tekst (Stream Editor)
sort	Sorteer tekst
tail	Toon de laatste regels van een tekstbestand
tr	Zoek en vervang lettertekens in tekst
uniq	Verwijder dubbele rijen uit een gesorteerd tekstbestand
wc	Tel karakters, woorden of lijnen in een tekstbestand

Filters: voorbeelden

```
# Invoer uit bestand
grep 'Williams' tennis.txt
sort -k2 tennis.txt
```

```
# Invoer via stdin
cat tennis.txt | grep 'Williams'
cat tennis.txt | tr 'a-z' 'A-Z'
```

```
# Combinatie
sort music.txt | uniq
```

Sed: voorbeelden

```
# Zoeken en vervangen (1x per regel)
sed 's/foo/bar/'

# "Globaal", meerdere keren per regel
sed 's/foo/bar/g'

# Regels die beginnen met '#' verwijderen
sed '/^#/d'

# Lege regels verwijderen
sed '/^$/d'
```

Awk: voorbeelden

Wat tussen accolades staat wordt uitgevoerd op elke regel

```
# Druk 4e kolom af (afgebakend door "whitespace")
awk '{ print $4 }'
```

```
# Enkel regels afdrukken die beginnen met #
awk '/^#/ { print $0 }'
```

```
# Druk kolom 2 en 4 af, gescheiden door ;
awk '{ printf "%s;%s", $2, $4 }'
```

Hoofdstuk 4. Scripting: intro

Vooraf: Git configuratie (1)

1 - Configureer Git in je VM

```
sudo dnf install git
git config --global user.name "Voornaam Naam"
git config --global user.email "emailadres@student.hogent.be"
git config --global push.default simple
git config --global pull.rebase true
git config --global core.autocrlf=input
```

Vooraf: Git configuratie (2)

2 - Maak een sleutelpaar

```
ssh-keygen
```

3 - Registreer publieke sleutel op Github

```
cat ~/.ssh/id_rsa.pub
```

4 - Kloon Git-repo in je VM

```
git clone git@github.com:HoGentTIN/linux-1819-USERNAME.git
```

met USERNAME jouw Github gebruikersnaam

Een script schrijven

1. Maak bestand aan (bv. script.sh) met een teksteditor, bv.

```
#!/bin/bash
echo "Hallo ${USER}"
```

2. Maak bestand uitvoerbaar: `chmod +x script.sh`

3. Voer uit: `./script.sh`

De “shebang”

- Eerste regel van een script
- Begint met `#!` (`#` = hash; `!` = bang)

- Absoluut pad naar de interpreter voor het script, bv:
 - `#!/usr/bin/python`
 - `#!/usr/bin/ruby`
 - `#!/usr/bin/env bash` (zoek in `${PATH}` naar bash)

Tekst afdrukken

Wat is het verschil?

```
var="world"
```

```
echo "Hello ${var}"
echo 'Hello ${var}'
```

```
var="world"
```

```
echo "Hello ${var}"      # Binnen " " wordt substitutie toegepast
echo 'Hello ${var}'      # Binnen ' ' NIET!
```

Gebruik printf

printf is beter dan echo

```
var="world"
```

```
printf 'Hello %s\n' "${var}"
```

- Het gedrag is beter gedefinieerd over verschillende UNIX-varianten.
- Vgl. `printf()` method in Java!

Fouten opsporen (1)

- Werk altijd **stap voor stap**
- **Test** voortdurend het resultaat van elke wijziging
- Hou minstens **2 vensters** open naast elkaar:
 - Editor
 - Terminal voor testen

Fouten opsporen (2)

- Syntax check: `bash -n script.sh`
- ShellCheck: `shellcheck script.sh`
 - Gebruik editor-plugin waar mogelijk
- Druk veel info af (printf)
- Debug-mode:
 - `bash -x script.sh`
 - In het script: `set -x` en `set +x`

Fouten voorkomen

Begin elk script met:

```
set -o errexit    # abort on nonzero exitstatus
set -o nounset    # abort on unbound variable
set -o pipefail   # don't hide errors within pipes
```

Variabelen

Bash-variabelen zijn (meestal) strings.

- Declaratie: `variabele=waarde`
 - **geen** spaties rond =
- Waarde v/e variabele opvragen: `${variable}`
- Gebruik in strings (met dubbele aanhalingstekens):

```
echo "Hello ${USER}"
```

Variabelen: tips

- Gebruik zoveel mogelijk de notatie `${var}`
 - *accolades*
- Gebruik dubbele aanhalingstekens: `"${var}"`
 - vermijd *splitsen* van woorden

```
bestand="Mijn bestand.txt"
touch ${bestand}           # Fout
touch "${bestand}"         # Juist
```

-
- Onbestaande variabele wordt beschouwd als *lege string*.
 - Oorzaak van veel fouten in shell-scripts!
 - `set -o nounset` □ script afbreken

Zichtbaarheid variabelen (scope)

Enkel binnen zelfde “shell”, niet binnen “subshells”

- Een script oproepen creëert een subshell
- Maak “globale”, of *omgevingsvariabele* met `export`:

```
export VARIABLE1=value
```

```
VARIABLE2=value
export VARIABLE2
```

Conventie naamgeving:

- Lokale variabelen: kleine letters, bv: `foo_bar`
- Omgevingsvariabelen: hoofdletters, bv: `FOO_BAR`

Hoofdstuk 5. Gebruikers, groepen en permissies

Commando's voor gebruikers en groepen

- Gebruikers: `useradd`, `usermod`, `userdel`

- Groepen: groupadd, groupmod, groupdel
- Info opvragen: who, groups, id

Configuratiebestanden

- Gebruikers: /etc/passwd, /etc/shadow
- Groepen: /etc/group, (/etc/gshadow, van weinig belang)

Primaire vs aanvullende groepen

- Primaire groep: 4e veld van /etc/passwd (group ID)
- Aanvullende groepen: in /etc/group. Gebruiker staat niet vermeld in de primaire groep!

primaire groep aanpassen

`sudo usermod -g groep gebruiker`

gebruiker toevoegen aan opgegeven groepen en verwijderen uit alle andere groepen

`sudo usermod -G groep1,groep2 gebruiker`

gebruiker toevoegen aan opgegeven groep, blijft lid van andere groepen

`sudo usermod -aG groep gebruiker`

Bestandspermissies (1)

= toegangsrechten voor bestanden en directories

- Bestanden zijn eigendom van een gebruiker en groep
- cfr. `ls -l` voor een overzicht

Bestandspermissies (2)

Instelbaar op niveau van:

- u: gebruiker, **u**ser
- g: groep, **g**roup
- o: andere gebruikers, **o**thers
- a: iedereen, **a**ll

Soorten permissies

- r: lezen, **r**ead
- w: schrijven, **w**rite
- x: **e**xecute
 - bestand: uitvoeren als commando
 - directory: toegang met `cd`

Instellen met symbolische notatie

permissies instellen met `chmod`, symbolische notatie

```

chmod MODUS FILE
chmod u+r FILE
    g-w
    o=x
    a

```

Voorbeelden:

- `chmod g+rw bestand`
- `chmod +x bestand`
- `chmod u+rw,g+r,o-rw bestand`
- `chmod a=r bestand`

Instellen met octale notatie

u			g			o		
r	w	x	r	-	x	-	-	-
1	1	1	1	0	1	0	0	0
4+2+1			4+0+1			0+0+0		
7			5			0		

Voorbeelden:

- `chmod 755 script`
- `chmod 600 file`
- `chmod 644 file`
- `chmod 775 dir`

Merk op

- sommige permissie-combinaties komen nooit voor in de praktijk, bv. 1, 2, 3
- directories: altijd lezen (r) en execute (x) *samen* toekennen of afnemen
- permissies owner \geq group \geq others
- root negeert bestandspermissies (mag alles), vb. `/etc/shadow`
- tip: octale permissies opvragen: `stat -c %a BESTAND`

Permissies van nieuwe bestanden: umask

- umask bepaalt permissies van bestand/directory bij aanmaken
- huidige waarde opvragen: `umask` zonder opties
- opgegeven in octale notatie
 - enkel 0, 2 en 7 zijn zinvol
- welke permissies *afnemen*
 - bestand krijgt nooit execute-permissie

Voorbeeld umask

`umask 0027`, wat worden de permissies?

file	directory	
0 6 6 6	0 7 7 7	basis
- 0 0 2 7	- 0 0 2 7	umask
-----	-----	

0 6 4 0 0 7 5 0 permissies

Speciale permissies: **SUID**

- set user ID (*SUID*)
- op bestanden met execute-permissies
- tijdens uitvoeren krijgt de gebruiker de rechten van de eigenaar van het bestand
- symbolisch: u+s
- octaal: 4

```
$ ls -l /bin/passwd
-rwsr-xr-x. 1 root root 28k 2017-02-11 12:02 /bin/passwd
```

Speciale permissies: **SGID**

- set group ID (*SGID*)
- op bestanden met execute-permissies
- tijdens uitvoeren krijgt de gebruiker de rechten van de groep van het bestand
- symbolisch: g+s
- octaal: 2

```
$ ls -l /usr/bin/write
-rwxr-sr-x. 1 root tty 20k 2017-09-22 10:55 /usr/bin/write
```

Speciale permissies: **restricted deletion**

- restricted deletion, of *sticky bit*
- toegepast op directories
- een bestand mag in zo'n directory enkel door de eigenaar verwijderd worden
- symbolisch: +t
- octaal: 1

```
ls -ld /tmp
drwxrwxrwt. 16 root root 360 2017-12-04 13:05 /tmp/
```

Eigenaar/groep veranderen:

Merk op: root-rechten nodig

```
chown user file
chown user:group file
chgrp group file
```

Hoofdstuk 6. Scripts schrijven

Positionele parameters

Bij uitvoeren van script zijn opties en argumenten beschikbaar via variabelen, *positionele parameters*

Variabele	Betekenis
<code>\${0}</code>	Naam script
<code>\${1}</code> , <code>\${2}</code> , ...	Eerste, tweede, ... argument

Variabele	Betekenis
<code>\${10}</code>	Tiende argument (accolades verplicht!)
<code>\${*}</code>	Alle argumenten: <code>\${1}</code> <code>\${2}</code> <code>\${3}</code> ...
<code>\${@}</code>	Alle argumenten: <code>"\${1}"</code> <code>"\${2}"</code> <code>"\${3}"</code> ...
<code>\${#}</code>	Aantal positionele parameters

Positionele parameters: shift

Het commando `shift` schuift positionele parameters op naar links:

- `${1}` verdwijnt
- `${2}` wordt `${1}`
- `${3}` wordt `${2}`
- enz.

Positionele parameters instellen

```
set par1 par2 par3
echo "${1}" # => par1
echo "${2}" # => par2
echo "${3}" # => par3
echo "${4}" # => (lege string)
```

Exit-status

- Elk commando heeft een *exit-status*, numerieke waarde
 - Opvragen met `echo "$?"`
 - 0 => commando geslaagd, logische *true*
 - 1-255 => commando gefaald, logische *false*
- Logische operatoren in Bash zijn gebaseerd op exit-status
- Booleaanse variabelen *bestaan niet*

Logische operatoren

```
if COMMAND0; then
  # A
else
  # B
fi
```

- A-blok wordt uitgevoerd als exit-status van `COMMAND0` 0 is (geslaagd, *TRUE*)
- B-blok wordt uitgevoerd als exit-status van `COMMAND0` verschillend is van 0 (gefaald, *FALSE*)

Het commando test

- Evalueren van logische expressies
- Geeft geschikte exit-status
 - 0 = *TRUE*
 - 1 = *FALSE*
- Alias voor `test` is `[`
 - `[` is een *commando*, geen “haakje” in de traditionele betekenis
 - spaties vóór en na!

```
# Fout:
if[$#-eq 0]; then
    echo "Expected at least one argument"
fi
```

```
# Juist:
if [ "${#}" -eq "0" ]; then
    echo "Expected at least one argument"
fi
```

If (1)

```
if EXPR
then
    # ...
elif EXPR
    # ...
else
    # ...
fi
```

If (2)

```
if [ "${#}" -gt '2' ]; then
    printf 'Expected at most 2 arguments, got %d\n' "${#}" >&2
    exit 1
fi
```

While-lus

```
while EXPR; do
    # ...
done
```

Until-lus

```
until EXPR; do
    # ...
done
```

While lus met teller

```
counter=0

while [ "${counter}" -le '10' ]; do
    echo "${counter}"
    counter=$((counter + 1))
done
```

`$((...))` is *arithmetic expansion*

For-lus

Itereren over een lijst

```
for ITEM in LIST; do
    # ...
done

for file in *.md; do
    printf 'Processing file %s\n' "${file}"
    # ...
done
```

For-lus met teller (1/3)

```
for i in {1..10}; do
    echo "${i}"
done

for i in {2..20..2}; do
    echo "${i}"
done
```

For-lus met teller (2/3)

```
for i in $(seq 1 10); do
    echo "${i}"
done

for i in $(seq 2 2 20); do
    echo "${i}"
done
```

For-lus met teller (3/3)

“Klassieke” for-lus, komt slechts uitzonderlijk voor

```
for ((i=0; i<=10; i++)); do
    echo "${i}"
done
```

Itereren over positionele parameters (while)

```
while [ "$#" -gt 0 ]; do
    printf 'Arg: %s\n' "${1}"
    # ...
    shift
done
```

Itereren over positionele parameters (for)

```
for arg in "${@}"; do
    printf 'Arg: %s\n' "${arg}"
    # ...
done
```

Hoofdstuk 7. Een webserver installeren

Doelstelling

- LAMP-stack: **L**inux + **A**pache + **M**ariaDB + **P**HP
- Installeren Wordpress
- Website bekijken vanop het fysieke systeem

Netwerkinstellingen controleren

Om Internettoegang mogelijk te maken zijn er 3 instellingen nodig:

1. IP-adres en subnetmasker
2. Default gateway
3. DNS-server

Netwerkinstellingen opvragen

1. IP-adres/netmask: `ip address (ip a)`
2. Default gateway: `ip route (ip r)`
3. DNS-server: `cat /etc/resolv.conf`

Wat is het IP-adres van...?

```
$ nslookup www.hogent.be
$ dig www.hogent.be
```

Wat is *mijn publiek* IP-adres?

```
$ curl icanhazip.com
81.164.175.191
```

Controleer eerst netwerkinstellingen

```
$ ip -4 a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group def
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 74751sec preferred_lft 74751sec
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group def
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s8
        valid_lft 1049sec preferred_lft 1049sec
```

Netwerkinstellingen

- lo (loopback): 127.0.0.1/8
- enp0s3 (VirtualBox NAT interface): 10.0.2.15/24
- enp0s8 (VirtualBox Host-only Adapter): 192.168.56.101/24

Problemen oplossen

- Geen enp0s8 of geen/verkeerd IP-adres op enp0s8?
- Volg instructies installatie VirtualBox:
 - Sluit VM af
 - Studiewijzer Besturingssystemen, §6.1, VirtualBox Configuratie
 - Hoofdvenster VirtualBox > VM > Details > Network
 - Maak 2e adapter aan, sluit aan op Host-only netwerk
 - Start VM op

Installatie software

```
$ sudo dnf install mariadb-server wordpress phpMyAdmin
```

Belangrijke directories

- /etc/httpd/: configuratie Apache
 - /etc/httpd/conf/httpd.conf
 - /etc/httpd/conf.d/*.conf
- /var/www/html/: Apache DocumentRoot
- /var/log/httpd/: logbestanden
 - access_log
 - error_log

Services opstarten

```
$ sudo systemctl start mariadb
$ sudo systemctl enable mariadb
$ sudo systemctl start httpd
$ sudo systemctl enable httpd
```

- start: *nu* opstarten
- enable: automatisch opstarten bij *booten*

Test de services

```
$ systemctl status httpd
$ systemctl status mariadb
```

- Open webbrowser *op de VM*
 - surf naar http://localhost/
- PHP testen: maak bestand /var/www/html/info.php
- Surf naar: http://localhost/info.php

```
<?php phpinfo(); ?>
```

Logbestanden

Voorbeeld voor Apache:

```
$ sudo journalctl
$ sudo journalctl -u httpd
$ sudo journalctl -flu httpd
```

```
$ sudo tail -f /var/log/httpd/access_log
$ sudo tail -f /var/log/httpd/error_log
```

Website vanaf fysiek systeem bekijken

- Controleer IP-adres VM: `ip a`
 - waarschijnlijk 192.168.56.101
- Open webbrowser *op fysiek systeem*
 - surf naar `http://192.168.56.101/`

Routing VM ↔ fysiek systeem

- Fysiek systeem → VM: `ping 192.168.56.101`
- VM → fysiek systeem: `ping 192.168.56.1`

Controleer instellingen Host-Only Netwerkinterface VirtualBox.

Firewall-instellingen aanpassen

```
$ sudo firewall-cmd --list-all # = toon firewall-regels
$ sudo firewall-cmd --add-service http --permanent
$ sudo firewall-cmd --add-service https --permanent
$ sudo firewall-cmd --reload
```

Probeer opnieuw de website te bekijken vanaf het fysieke systeem

Database beveiligen

```
$ sudo mysql_secure_installation
```

- Volg de instructies!
- kies MariaDB root-wachtwoord
 - ≠ wachtwoord Linux root!
- bevestig andere vragen (ENTER)

Hou je wachtwoorden goed bij!

Database voor Wordpress aanmaken

- Surf naar `http://localhost/phpmyadmin/`, log in
- Klik op “User accounts” → “Add user account”
 - User name: `wordpress`
 - Host name: `Local / localhost`
 - Password: kies er een of “Generate”
- Create database with same name and grant all privileges: *aanvinken*
- Ga naar onderaan de pagina, klik rechtsonder op “Go”

Database testen: root

```
$ mysql -uroot -pR2rrbLV02TA1hAjN mysql
...
MariaDB [mysql]> SHOW DATABASES;
MariaDB [mysql]> SELECT user,password from user;
MariaDB [mysql]> quit
```

- -uroot: inloggen als MariaDB-root
- -pR2rrbLV02TA1hAjN: gekozen wachtwoord
 - (GEEN spatie na -p)
- mysql: inloggen op database mysql

Database testen: wordpress

```
$ mysql -uwordpress -pPu7QxGKPlvEdw6Gr wordpress
...
MariaDB [wordpress]> SHOW TABLES;
MariaDB [wordpress]> quit
```

Wordpress: Belangrijke bestanden

- /usr/share/wordpress/: installatie Wordpress
- /etc/wordpress/wp-config.php: configuratie
- /etc/httpd/conf.d/wordpress.conf: Apache-configuratie

Wordpress: Apache-configuratie

In /etc/httpd/conf.d/wordpress.conf:

```
# Zoek naar
Require local
# Vervang door
Require all granted
```

Daarna: sudo systemctl restart httpd

Wordpress: database-instellingen

In /etc/wordpress/wp-config.php:

```
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'wordpress');

/** MySQL database password */
define('DB_PASSWORD', 'Pu7QxGKPlvEdw6Gr');
```

Wordpress: Keys en Salts

- Ga naar <https://api.wordpress.org/secret-key/1.1/salt/>
- Copy-paste naar wp-config.php

Wordpress: installatie plugins en updates

In /etc/wordpress/wp-config.php:

```
/* ENABLE file changes */
define('DISALLOW_FILE_MODS', false);
```


Installatie Wordpress

- Surf naar <http://192.168.56.101/wordpress/>
- Kies titel, gebruikersnaam, wachtwoord, enz.
- Log in
- Klaar!

Hoofdstuk 8. Complexere scripts

Case (1)

```
case EXPR in
  PATROON1)
    # ...
    ;;
  PATROON2)
    # ...
    ;;
  *)
    # ...
    ;;
esac
```

Case (2)

```
option="${1}"

case "${option}" in
  -h|--help|-?)
    usage
    exit 0
    ;;
  -v|--verbose)
    verbose=y
    shift
    ;;
  *)
    printf 'Unrecognized option: %s\n' "${option}"
    usage
    exit 1
    ;;
esac
```

Communicatie script/omgeving

Informatie uitwisselen tussen script en omgeving:

- I/O: stdin, stdout, stderr
- Positionele parameters: \$1, \$2, enz.
- Exit-status (0-255)
- Omgevingsvariabelen, vb:

```
VAGRANT_LOG=debug vagrant up
```

Functies in Bash

```
functie_naam() {  
    # code  
}
```

Een functie gedraagt zich als een script!

- oproepen: `functie_naam arg1 arg2 arg3`
- positionele parameters: `${1}`, `${2}`, enz.
- return `STATUS` ipv `exit`

Scope variabelen bij functies

Wat is de uitvoer van dit script?

```
#!/usr/bin/env bash  
var_a=a  
  
foo() {  
    var_b=b  
    echo "${var_a} ${var_b}"  
}  
  
foo  
  
echo "${var_a} ${var_b}"
```

Functies in Bash: voorbeeld

```
# Usage: copy_iso_to_usb ISO_FILE DEVICE  
# Copy an ISO file to a USB device, showing progress with pv (pipe viewer)  
# e.g. copy_iso_to_usb FedoraWorkstation.iso /dev/sdc  
copy_iso_to_usb() {  
    local iso="${1}"  
    local destination="${2}"  
    local iso_size  
  
    iso_size=$(stat -c '%s' "${iso}")  
  
    printf "Copying %s (%'dB) to %s\n" \  
        "${iso}" "${iso_size}" "${destination}"  
  
    dd if="${iso}" \  
        | pv --size "${iso_size}" \  
        | sudo dd of="${destination}"  
}
```

Tips (I)

- Begin elk script met `set -o ...`
- Zet positionele parameters om in beschrijvende namen
- Maak lijnen niet te lang (gebruik `\` op het einde van een regel)
- Gebruik “lange” opties: maakt script leesbaarder

```
set -o nounset # stop bij onbestaande variabele
set -o errexit # stop bij fout (exitstatus ≠ 0) in commando
set -o pipefail # stop bij fout in pipeline
```

Tips (2)

- Gebruik lokale variabelen in functies
- Deel script op in (herbruikbare) functies

Meer tips op: <https://gitpitch.com/bertvv/presentation-clean-bash>