## Linux Aanvullende Oefeningen

# Hoofdstuk 5 en 6

Installeer het tree-commando

**sudo apt-get install tree**

Zoek in de manpages hoe je met het tree-commando ENKEL de directories vanaf / kan zien en maar één niveau omlaag.

**tree / -L 1**

Waarom werkt tab-completion niet als je het volgende doet : vi /home/student/.bash<tab>

**Omdat er meerdere bestanden zijn die met .bash beginnen**

**Nogmaals op <tab> drukken geeft een lijst met alle files die met .bash beginnen**

Zoek via de manpages hoe je met ls een overzicht kan krijgen

*   van je homedir inclusief de verborgen items
* **ls -a**
*   geordend op de bestandsgrootte
* **ls -S**
*   van klein naar groot
* **ls -Sr**
*   met de bestandsgroottes respectievelijk in  bytes, kilobytes, megabytes, gigabytes
* **ls -Srh**
* Maak volgende boomstructuur aan: /home/student/school/aca\_2012\_2013/trimesters/2/vakken/ubuntuserver/oefeningen/chapter5
* **mkdir -p ~/school/aca\_2012\_2013/trimesters/2/vakken/ubuntuserver/oefeningen/chapter5**
* Maak, in de folder oefeningen, ook een chapter6 –folder aan
* **mkdir -p ~/school/aca\_2012\_2013/trimesters/2/vakken/ubuntuserver/oefeningen/chapter6**
* Gooi chapter5 weg
* rmdir ~**/school/aca\_2012\_2013/trimesters/2/vakken/ubuntuserver/oefeningen/chapter5**
* Gooi alle folders vanaf school weg
* **rm -rf ~/school/aca\_2012\_2013/**
* Zoek via de manpage van man wat het de optie -f (man -f) precies doet en door welk ander commando het vervangen kan worden
* **Display a short description of a manual**
* **Kan vervangen worden door --whatis**
* Zoek een manpage, die bestanden met de extensie gz, kan unzippen. Kopieer de manpage van whoami naar je homedir, unzip en bekijk de inhoud.
* **man -k gz // zoekt naar manpage**
* **man -wa whoami // geeft pad naar manpage**
* **cp /usr/share/man/man1/whoami.1.gz ~/ // kopieer de manpage naar de homedir**
* **cd // ga naar homedir**
* **gunzip whoami.1.gz //unzip met gunzip**
* **cat whoami.1 // bekijk de inhoud**
* Ga naar de directory waar de home-page van je website staat.
* **cd /var/www/html**

Ga naar je homedir

**cd**

cat in één commando de index-file van je website door gebruik te maken van een RELATIEF pad !

**cat ../../var/www/html/index.html**

Probeer met maximum 5 toetsaanslagen opnieuw naar de map te gaan waar de home-page van website staat. Geef 2 oplossingen.

**cd -**

**!cd /**

Maak het scherm leeg.

**clear**

Wat zijn de verschillen tussen de volgende commando’s ls /etc/\*

ls /etc/\*.\* ls /etc/\*a.\*

**ls /etc/\*.\* geeft alle bestanden in /etc weer**

**ls /etc/\*a.\* geeft alle bestanden in /etc weer waar de naam eindigt op a (extensie exclusief)**

Zorg dat je je bevindt in je homedirectory. Kopieer de volledige directory naar de subdirectory kopij in je eigen homedirectory

**cp -r \* kopij**

Extra challenge: Hoe gebruik je het commando ls om alle bestanden in de directory /etc te tonen waar op de tweede positie van de bestandsnaam een a, o of i voorkomt?

**ls ?[a,o,i]\*.\***

# Hoofdstuk 7 en 8

1. Ga naar je homedir

**cd**

2. Maak, in je homefolder, op een makkelijke en snelle manier een leeg bestandje aan, genaamd “touchtest”.

**touch touchtest**

3. Maak met één commando volgende mappenstructuur aan in je homefolder folder/subfolder/subsubfolder

**mkdir -p folder/subfolder/subsubfolder**

4. Maak, zonder uit je homefolder te gaan, een bestand aan in iedere map.

**touch folder/file1 folder/subfolder/file2 folder/subfolder/subsubfolder/file3**

5. Verwijder met één commando de map folder met alle submappen.

**rm -rf folder**

6. Maak een nieuwe map fotos. Maak, met slechts één commando, in de map fotos de volgende bestanden aan

Foto1.JPG, foto2.JPG, Foto3.jpg, foto4.jpg (let op de hoofdletters)

**mkdir fotos**

**touch /folder/Foto1.JPG fotos/foto2.JPG fotos/Foto3.jpg fotos/foto4.jpg**

7. Hernoem de bestanden zodanig dat alles in kleine letters komt te staan.

**rename 'y/[A-Z]/[a-z]/' \***

8. Zoek uit waarom je beter less gebruikt dan more.

**more is een primitief commando, less heeft meer uitgebreide verbeteringen**

9. Maak een bestand aan, genaamd “test.c”, met onderstaande tekst.

**cat > test.c**

#include <stdio.h> int main(){

char naam[30]; printf( “Hoe heet je?:”); scanf(“%s”,naam); printf(“Hallo %s.\n”, naam); getchar(); getchar(); return 0;

**control-c**

} 10. Compileer het bestand met de C-compiler, met onderstaand commando

**cc test.c –o test.out**

11. Test het programma met volgend commando: ./test.out 12. Zoek naar de leesbare ascii-tekens in de binary-file (test.out)

**strings test.out**

# Hoofdstuk 9 en 10

Zoek uit of cd een Builtin-commando is. Zoek uit of mkdir een Built-in commando is.

**type cd**

Indien we ll gebruiken om de inhoud van een directory te zien, welk commando wordt er dan precies uitgevoerd?

**alias ll**

**antwoord -> alias ll='ls -alF'**

Geef het commando dat kan aantonen dat het kill-commando tegelijk builtin en external is.

**type -a kill**

Maak een alias voor dir, zodanig dat deze een “ls -lah” uitvoert. Doe een dir van je homedir

**alias dir='ls -lah'**

Voeg een aantal regels toe aan het bestand .profile van je homedir, zodanig dat in de toekomst, indien nodig, voor bevestiging wordt gevraagd bij de de commando's rm, mv en cp.

**echo "alias rm='rm -i'\nalias mv='mv -i'\nalias cp='cp -i'" >> .profile**

Zorg er voor dat dit ook automatisch in het profiel komt van alle nieuwe aangemaakt gebruikers van de pc.

**sudo cp .profile /etc/skel/**

Herstart de VM Toon alle aliases

**alias**

Voeg een regel toe aan het bestand .profile van je homedir, zodanig dat in de toekomst het volgend commando bestaat

dirs => gebruikt het commando tree => toont alle directories, ook de verborgen

=> geen subdirectories, dus maar één niveau => geen indentatie-tekens

**echo "alias dir='tree -a -i -L 1'" >> .profile**

Probeer met slechts één echo-commando de volgende drie cijfers telkens op een aparte regel te printen. Een Twee Drie

**echo -e "Een\nTwee\nDrie"**

# Hoofdstuk 11 en 12 en 13

Maak een commando dat bij het verwijderen van de file “test” :

‒  de gewone foutmeldingen onderdrukt door ze naar het zwarte gat te sturen ( 2>...)

‒  de tekst “succesvol verwijderd!” teruggeeft, indien hij het bestand heeft kunnen verwijderen

‒  de tekst “fout tijdens het verwijderen” teruggeeft, indien hij het bestand niet heeft kunnen verwijderen

**rm test 2> /dev/null && echo "Succesvol verwijderd!" || echo "Fout tijdens het verwijderen"**

Zorg dat de prompt er als volgt uit ziet. Gebruik man bash en zoek naar PROMPT voor de opties.

<lege lijn>

*Fri Jan 18 10:24:45 – PTS4*

*student@ubuntu:~$ ls*

*dir1 dir2 bestand1*

*<lege lijn>*

*Fri Jan 18 10:24:59 – PTS4*

*student@ubuntu:~$*

***PS1="\n\d \t - PTS4\n\u@\h:~$ "***

Zorg dat deze prompt altijd wordt gebruikt

**echo "*PS1=\"\n\d \t - PTS4\n\u@\h:~$ \"" >> .bashrc***

Maak een commando dat de volgende tekst met de **huidige** datum op het scherm afdrukt. (Tip: Maak gebruik van meerdere variabelen of van shell embedding.)  => Today, it is monday, 21 january 2013. The time is 15:30.

**echo "Today, it is $(date +'%A, %d %B %Y'). The time is $(date +%R)."**

Zorg ervoor dat scriptjes die in de huidige directory staan, ook steeds gevonden worden. Je dient dus geen scriptjes meer uit te voeren met ./script.sh. (tip: /etc/init.d/rc.local)

**sudo vim /etc/init.d/rc.local // edit het bestand in vim**

**:set number // zet lijnnummering aan**

**12 PATH=sbin:/usr/sbin:/bin:/usr/bin**

wordt

**12 PATH=sbin:/usr/sbin:/bin:/usr/bin:/bin:/~/**

**ESC**

**:wq**

Gebruik een commando om al de variabelen die ingesteld zijn te zien.

**set**

Plaats in een variable, genaamd “mijnvar” de tekst “super”. Druk de volgende tekst af op het scherm, gebruikmakende van de variable. *Dag supervrouw, ik ben superman !*

**mijnvar="super"**

**echo "Dag $(echo $mijnvar)vrouw, ik ben $(echo $mijnvar)man !"**

Verwijder de variable en druk de tekst opnieuw af.

**unset mijnvar**

output -> **Dag vrouw, ik ben man !**

Zorg ervoor dat volgende tekst op het scherm wordt afgedrukt en voor iedere gebruiker correct is. (Zorg dat ook de hidden files worden meegeteld.) => In mijn homefolder zitten 15 files en folders.

**echo "In mijn homefolder zitten $(ls -p ~/ | grep -v '/$' | wc -l) files en $(ls -d ~/\*/ | wc -l) folders."**

# Hoofdstuk 14 en 15

Maak de bestanden script.sh, scrupt.sh en scrApt.sh aan. Gebruik hiervoor driemaal het touch commando.

**touch script.sh**

**touch scrupt.sh**

**touch scrApt.sh**

Type CTRL-R om te zoeken in de bash\_history en type enkel de letters scr. Je ziet nu het laatste commando dat je gebruikt hebt dat hieraan voldoet, dus normaal gezien “touch scrApt.sh”.

Hoe kan je, zonder letters bij te typen, toch naar het commando touch “script.sh”?

**Nogmaals CTRL-R indrukken tot je het commando ziet dat je wil uitvoeren**

Zorg ervoor dat het volgende commando alle drie de sh-scriptjes laat zien: ls scr[a-z]pt.\*

**ls scr[a-z]pt.\***

Zorg er voor dat uit de history de laatste touch wordt aangepast, zodanig dat de letter “i” wordt vervangen door de letter “u”. Gebruik hiervoor regular expressions.

**history**

**!n:s/i/u**

Maak met één commando volgende bestanden aan: script.sd script.dh script.ds

**touch script.sd script.dh script.ds**

Zoek in je folder naar alle bestanden die eindigen op .sh of .sd of .dh of .ds

**ls -l \*.[sd][hds]**

Zoek in je folder naar alle bestanden die beginnen met scr, dan eender welk teken en vervolgens de letters pt. De files moeten ook eindigen met een h of een s.

**ls -l scr?pt.\*[hs]**

# Hoofdstuk 16, 17 en 18

Wat doet volgend commando ? > test

**Maakt het bestand "test" aan**

Zorg er voor dat je de vijf laatst aangemaakte user-accounts in een overzichtje krijgt, maar dan geordend van de laatst aangemaakte (= meest recent toegevoegde) tot de eerst aangemaakte (=oudste).

**cat /etc/passwd | tail -5 | cut -d':' -f1| tac**

Geef de optie om met grep niet hoofdlettergevoelig te zoeken.

**grep -i**

Zorg er voor dat volgende tekst wordt onderzocht op je computer en als volgt afgebeeld.

De computer ubuntu heeft momenteel het ip-adres 172.16.12.128 verkregen via zijn mac-adres 000C29DDED7C.

*Let op: het mac-adres heeft geen dubbele punten en de letters staan in hoofdletters.*

Doe dit alles in één commando(regel). Maak dus GEEN gebruik van “;” om meerdere commando's achter elkaar uit te voeren op één commandoregel ! *ubuntu → dit is de hostname van de computer 172.16.12.128 → is het ip-adres (tip: ifconfig eth0) 000C29DDED7C → mac-adres (tip: ifconfig eth0)*

**echo -e "$(echo $HOSTNAME) -> Dit is de hostname van de computer $(ifconfig | head -2 | tail -1 | cut -d':' -f2 | tr -d ' Bcast')\n$(ifconfig | head -1 | cut-d':' -f2,3,4,5,6,7 | tr -d 'Ethernet HWaddr ' | tr-d ':' | tr '[a-z] '[A-Z]') -> mac-adres**

Toon de lijnen en lijnnummers uit het bestand linux.txt die het woord minix of MINIX bevatten.

Voorbeeld linux.txt:

Deze lijn bevat Linux

Deze lijn bevat Minix

Deze lijn bevat Unix

Unix

Minix

Unix

Test

Test

Test

**cat linux.txt | grep -nrE 'minix|MINIX'**

Toon de lijnen van het bestand linux.txt die de woorden Linux, Minix of Unix bevatten

**cat linux.txt | grep -E 'Linux|Minix|Unix'**

Hoeveel lijnen uit het bestand linux.txt bevatten het woord Unix ?

**cat linux.txt | grep -E 'Unix' | wc -l**

Toon alle lijnen uit bestand linux.txt die beginnen met een hoofdletter

**cat linux.txt | grep -E '^[A-Z]\*'**

Tel het aantal users die op de ubuntu-vm kunnen inloggen

**cat /etc/passwd | grep -v 'nologin' | wc -l**

Druk de password-file af, geordend op de userid

**cat /etc/passwd | sort**

Geef de namen van alle users die minstens één proces hebben lopen, geordend op wie het meeste processen heeft lopen (ps aux)

**ps aux | cut -d ' ' -f1,12 | sort -V | cut -d ' ' -f1 | uniq**

Plaats een ls van je homedir in file1 en een ls -a van je homedir in file2. Geef een vergelijkend overzicht van welke files en folders enkel in file1, enkel in file2 of in beide files voorkomen.

**ls ~/ > file1 ; ls -a ~/ > file2 ; comm file1 file2**

Geef enkel de files die in beide bestanden staan

**comm file1 file2 -1,2**

Start met het commando ls van de root-folder (/) en probeer te eindigen met een lijst van files en folders die gescheiden door een spatie achter elkaar worden opgesomd.

**ls / | tr -s ' ' | tr '\n' ' '**

Doe een ls -la van je homefolder en verander in het overzicht de verwijzing “.” door “. (this folder)” en “..” door “.. (parent folder)”. Verwijder ook de lijn die het aantal files en folders van de dir aangeeft. *TIP: indien je het $-teken gebruikt in een achteraan de zoekstring van de regular expression, dan wil dit zeggen dat de regel moet eindigen met die zoekstring*.

**ls -la | tr '.' '.(thisfolder)' | tr '..' '..(parent folder)' | sed 1d**

Geef een overzicht van op hoeveel tty's en pts-en (voor ubuntu- desktop) iemand is ingelogd

**w | grep -E $USER | wc -l**

Ga naar je homedirectory. Plak alle files uit deze directory achter elkaar en tel vervolgens het aantal woorden.

**cd ; find . -type f -exec cat {} + | wc -w**

Maak een file aan in je homedir, genaamd nieuwbestand. Zorg ervoor dat je dit bestand kan vinden met het commando locate.

**touch nieuwbestand ; sudo updatedb ; locate nieuwbestand**

Plaats een overzicht van alle bestanden en mappen vanaf de root-folder in het bestand “overzicht”. Zorg er ook voor dat je na afloop weet hoe lang dit commando er over gedaan heeft.

**time ls -la / > ~/overzicht**

Bekijk de grootte van het bestand “overzicht” en dit best in human- readable-format.

**ls overzicht -lh**

Comprimeer dit bestand met de twee commando's die we hiervoor hebben gezien en bekijk het verschil in grootte van de twee bestanden. Time ook deze twee commando's om het bestand te comprimeren.

**time gzip -c overzicht > overzicht.gz ; time bzip2 overzicht**

Probeer de inhoud van de files die je gezipped hebt te bekijken zonder ze eerst uit te pakken. Zorg er voor dat je per scherm kan doorscrollen.

**bzcat overzicht.bz2 ; zcat overzicht.gz | less**

Decomprimeer het bestand opnieuw

**gunzip overzicht.gz ; bunzip2 overzicht.bz2**

Doe een find van de folder “/proc/” en zorg er voor dat de output van het commando terecht komt in een bestand genaamd “find\_results”. Tevens moeten de foutmeldingen, met uitzondering van de foutmelding met in de tekst het woord “task” opgeslaan worden in een bestand genaamd “find\_errors”. Doe dit alles in één commando(regel). Maak dus GEEN gebruik van “;” om meerdere commando's achter elkaar uit te voeren op één commandoregel !

**find /proc/ > find\_results 2> >(grep -v 'task' 2> find\_errors)**

Doe een find van de folder “/proc/” en zorg er voor dat je zowel de output van het commando, alsook de foutmeldingen, met uitzondering van de foutmelding met het woord “task”, opslaat in een bestand genaamd “find\_errors\_and\_results”. Doe dit alles in één commando(regel). Maak dus GEEN gebruik van “;” om meerdere commando's achter elkaar uit te voeren op één commandoregel !

**find /proc/ | grep -v 'task' | cat 2>&1 find\_errors\_and\_results**

Hoe zorg je dat je met volgend commando geen foutmeldingen 'Permission denied' meer krijgt:

find / -name output.txt

**sudo find / -name output.txt**

Maak volgend bestand aan en noem dit bestand ‘positie’:

Op de vierde positie bevindt zich een d

Op de tiende positie bevindt zich een n

Mijn auto kost 12000$

a)  Geef het commando dat alle regels toont met op de 4de positie een d

**cat positie | grep -E '^...d'**

b)  Geef het commando dat alle regels toont met op de 10de positie een n

**cat positie | grep -E '^.........n'**

c)  Geef het commando dat alle regels toont met een $ als laatste karakter

**cat positie | grep -E '\$$'**

Maak volgende bestanden aan:

**produkten**

rok:700

broek:400

schoenen:100

das:600

sokken:200

muts:800

trui:300

hemd:500

**prijzen**

400:€120

700:€80

300:€120

800:€12

600:€20

100:€100

500:€90

200:€5

Gebruik de 2 voorgaande bestanden om volgend bestand ‘**product\_prijs**’ te bekomen *(Tip: join)*

100:schoenen:€100

200:sokken:€5

300:trui:€120

400:broek:€120

500:hemd:€90

600:das:€20

700:rok:€80

800:muts:€12

**cat producten.txt | tr ':' ' ' | sort -t ' ' -k2 > tempproducten.txt**

**cat prijzen.txt | tr ':' ' ' | sort -t ' ' -k1 > tempprijzen.txt**

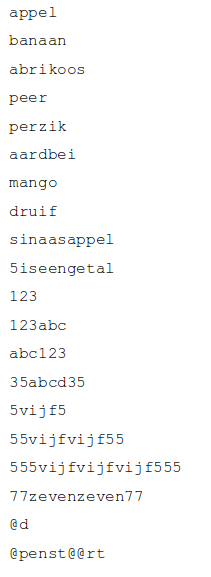
**join -1 2 -2 1 poducten.txt prijzen.txt > tempproduct\_prijs.txt**

**cat tempproduct\_prijs.txt | tr ' ' ':' > product\_prijs.txt**

**rm -rf tempproducten.txt tempprijzen.txt tempproduct\_prijs.txt**

Hoofdstuk 19 en 20

Maak regtest.txt aan:



● Maak een regular expressie waarmee uit regtest.txt de lijnen gaat

filteren die:

i. de letter a bevatten

**cat regtest.txt | grep 'a'**

ii. de letters a en s bevatten

**cat regtest.txt | grep '[as]'**

iii. een @ bevat

**cat regtest.txt | grep '@'**

iv. beginnen met de letter p

**cat regtest.txt | grep '^p'**

v. eindigen met de letter l

**cat regtest.txt | grep 'l$'**

vi. beginnen met een cijfer

**cat regtest.txt | grep '^[0-9]'**

vii. beginnen met een cijfer en eindigen met een letter

**cat regtest.txt | grep '^[0-9].\*[a-z]$'**

viii. 2 of meer opeenvolgende letters a bevat

**cat regtest.txt | grep 'aa+'**

ix. enkel cijfers bevatten

**cat regtest.txt | grep -v '[a-Z]'**

x. enkel letters bevatten

**cat regtest.txt | grep -v '[0-9]'**

xi. een of meer cijfers, gevolgd door een of meer letters bevatten

**cat regtest.txt | grep -E '[0-9]+[a-z]+'**

xii. een of meer cijfers, gevolgd door een of meer letters en eindigen op een cijfer

**cat regtest.txt | grep -E '[0-9]+[a-Z]+.\*[0-9]$'**

● Maak gebruik van sed om:

i. van uit regtest.txt een nieuw bestand regtest\_5.txt te maken waarin alle cijfers 5 door de tekst 'vijf' worden vervangen

**cat regtest.txt | sed 's/5/vijf/g' > regtest\_5.txt**

ii. van uit regtest.txt een nieuw bestand regtest\_7.txt te maken waarin de tekst 'zeven' door het cijfer 7 wordt vervangen

**cat regtest.txt | sed 's/zeven/7/g' > regtest\_7.txt**

iii. van uit regtest.txt een nieuw bestand regtest\_ad.txt waar in je alle @ vervangt door de tekst \_ad\_

*!!! Voor extended regex gebruik moet je sed –re ‘s/extendedregex/tevervangen/g’ gebruiken De –r zorgt ervoor dat sed extended regexen gebruikt.*

**cat regtest.txt | sed 's/@/\_ad\_/g' > regtest\_ad.txt**

● Tel hoeveel keer de tekst 5 voorkomt in regtest.txt

**cat regtest.txt | grep -o '5' | wc -l // grep -o zal alle occurences op een nieuwe lijn plaatsen**

**// wc -l telt vervolgens alle lijnen**

● Vervang ( met sed) in regtest.txt 5 door vijf en tel op hoeveel lijnen de term vijf voorkomt. Dit alles in één opdracht.

**cat regtest.txt | sed 's/5/vijf/g'| grep -o 'vijf' | wc -l**

● Zoek in de manpage van bash naar de hoofding (tekst begint tegen de linkerkantlijn!) genaamd “REDIRECTION”. Is deze zoek-functie hoofdlettergevoelig?

**man bash**

**/**

**redirection**

of

**REDIRECTION**

**Deze werken allebei en deze functie is dus niet hoofdlettergevoelig.**

● Probeer met dezelfde regular-expression enkel de verborgen bestanden uit je homefolder te tonen.

**ls -a ~/ | grep -E '^\.'**

● Zorg dat er steeds getoond wordt in welke modus je je bevindt (commando of insert) in vi.

**:set showmode**

● Start vi op en voer de volgende opdrachten uit:

a. Typ de volgende tekst: *Dit is een zinloos tekstbestandje. Het maakt deel uit van de oefening over vi.*

**vim**

***i***

**Dit is een zinloos tekstbestandje**

**ESC**

b. Sla dit bestand op onder de naam dit.is.een.bestand. Doe dit zonder dat vi wordt afgesloten.

**:w dit.is.een.bestand**

c. Voeg de volgende regel aan de tekst toe:

*Did iz we reeen regel tekst. Med feel vauten deese ceer.*

***o***

**Did iz we reeen regel tekst. Med feel vauten deese ceer.**

**ESC**

**:w**

d. Corrigeer de fouten (verwijder/vervang de foute karakters in commando modus)

***r* // R = vervang karakter**

***x* // X = verwijder karakter**

e. Sla het bestand op onder de naam testfile.1 en sluit gelijktijdig vi af.

**ESC**

**:wq testfile.1**

f. Startviweerop

**vi**

g. Voer de volgende tekst in: *Ook dit is weer zo’n suffe regel voorbeeldtekst*

***i***

**Ook dit is weer zo'n suffe regel voorbeeldtekst**

h. Sla de nieuwe tekst op onder de naam testfile.1 waarbij het reeds bestaande testfile.1 overschreven wordt.

***:w! testfile.1***

● Maak een nieuwe file ‘oef2\_vi’ aan.

**vi oef2\_vi**

Geef volgende tekst in:

*What is the capital of New Zealand ?*

*Wellington*

*There was an old man who loved eating lettuce. Nobody took any*

*Notice till one day he turned into a lettuce. He finally got eaten by*

*A large white rabbit*

***i***

**What is the capital of New Zealand ?**

**Wellington**

**There was an old man who loved eating lettuce. Nobody took any**

**Notice till one day he turned into a lettuce. He finally got eaten by**

**A large white rabbit**

**ESC**

Positioneer de cursor op het eerste karakter van de tweede paragraaf.

Verwijder het woord waarop de cursor staat gepositioneerd.

**dw**

Verwijder de ganse zin

**dd**

Positioneer de cursor op het eerste karakter van het woord ‘white”.

Verwijder vanaf de positie van de cursor de rest van de lijn.

**d$**

Verplaats lijn 1 en 2 naar het einde van de file

**:1,2m $**

Plaats regelnummers

**:set number**

● Maak, met vi een bestand aan, genaamd “nummers”, en plaats onmiddellijk 100 regels er in met telkens de respectievelijke regelnummer (de eerste regel heeft de tekst 1, de tweede regel heeft

de tekst 2, enz...)

**:for i in range (1,100) | put ='tekst '.i | endfor**

●  Zorg er voor dat je in de toekomst steeds de regelnummers ziet in vim.

**vim ~/.vimrc**

***i***

**set number::**

**:wq**

●  Zorg er ook voor dat je initialen onmiddellijk worden omgezet in volledige voornaam en naam.

**:ab vn Voornaam Achternaam**

●  Zorg er ook voor dat je met F6 het bestand opslaat en vim afsluit.

**:map <F6> :wq! <Return>**

●  Zorg er voor dat alle users op het systeem de regelnummers zien in  vim.

**sudo vim /etc/vim/vimrc**

***G***

**set numbers**

**:wq**

●  Zorg er voor dat, in het bestand “nummers” de regels met de nummers van 30 tot 69 verwijderd zijn.

**:30,69d**

●  Zorg dat er tussen de regel met nummer 29 en regel met nummer 70 de tekst komt “Hieronder wordt ineens naar nummer 70 gesprongen”.

***o***

**"Hier wordt ineens naar nummer 70 gesprongen"**

**ESC**

●  Vervang in het bestand nummers ieder cijfer 5 door cijfer 9.

**:%s/5/9/g**

●  Open met vi een nieuw bestand, genaamd overzicht\_users.  - Voeg de tekst “Overzicht” toe. - Voeg daaronder enkel de usernames toe van het bestand /etc/passwd  (via een commando in vi) - Sla het bestand op.

**vi overzicht\_users**

***i***

**Overzicht**

**ESC**

**:r /etc/passwd**

**:wq**

# Hoofdstuk 21 en 22

● Schrijf een script oef1.sh dat volgende output genereerd:

student@desktop:~/bin$ ./oef1.sh Dit is mijn eerste script Vandaag: 10/21/14 User: student

*De datum is via het commando date gegenereerd. De user is de huidig ingelogde user.*

**#!/bin/bash --**

**echo "Dit is mijn eerste script"**

**echo "Vandaag: $(date +'%m/%d/%Y) User: $USER"**

● Schrijf een script oef2.sh dat volgende output genereerd:

student@desktop:~/bin$ ./oef2.sh Ik ben student Mijn homedir is /home/student Ik heb 70 entries in /home/student

*student in het voorbeeld is de huidig ingelogde user. /home/student is de homedirectory van de huidig ingelogde user. Het aantal entries moet je via commando’s genereren.*

**#!/bin/bash --**

**echo "Ik ben $USER"**

**echo "Mijn homedir is $(echo ~)"**

**echo "Ik heb $(ls ~ | wc -l) entries in $(echo ~)"**

● Schrijf een script oef3.sh dat om de seconde een lijn afdrukt.

Bvb.:

student@desktop:~/bin$ ./oef3.sh I am ALIVE ...

*Tip: sleep*

**#!/bin/bash --**

**echo "I"**

**sleep 1**

**echo "am"**

**sleep 1**

**echo "ALIVE ..."**

● Pas vorige oefening aan en druk voor elke lijn een nagemaakte prompt. Gebruik hiervoor een variabele, bvb. $fakeprompt Bvb.

*student@desktop:~/bin$ ./oef3b.sh*

*student@desktop:/home/student/bin$ I*

*student@desktop:/home/student/bin$ am*

*student@desktop:/home/student/bin$ ALIVE ...*

**fakeprompt="student@desktop:/home/student/bin$"**

**echo "$fakeprompt I"**

**sleep 1**

**echo "$fakeprompt am"**

**sleep 1**

**echo "$fakeprompt ALIVE ..."**

● Schrijf een script oef4.sh dat controleert of je de laatst toegevoegde

user bent. Mogelijke output: student@desktop:~/bin$ ./oef4.sh Ik ben student en ben niet de laatst toegevoegde user. De laatst toegevoegde user is testuser. student@desktop:~/bin$ su ­ testuser Password: testuser@desktop:~$ /home/student/bin/oef4.sh Ik ben testuser en ben de laatst toegevoegde user.

*Tip: /etc/passwd Gebruik variabelen !!*

**#!/bin/bash --**

**lastuser=$(cat /etc/passwd | tail -1 | cut -d':' -f1)**

**if ["$lastuser" = $USER]**

**then**

**echo "Ik ben $USER en ik ben de laatste toegevoegd user."**

**else**

**echo "Ik ben $USER en ik ben niet de laatst toegevoegde user.\nDe laatst toegevoegde user is $lastuser"**

**fi**

● Schrijf een script oef5.sh dat om de seconde aftelt van 10 t.e.m. 1 en

daarna de tekst “KAPOW” laat flikkeren in het rood.

*Je moet voor de ‘blink’ te zien dit uitvoeren in xterm i.p.v. in Terminal. Je vindt xterm terug via de Dash.* student@desktop:~/bin$ ./oef5.sh

10 9 8 7 6 5 4 3 2 1 KAPOW

*Tip: sleep*

*Tip: Laatste lijn in je script moet er als volgt uitzien:* echo ­e "\033[1;31;5;7mKAPOW\033[0m"

|  |  |
| --- | --- |
| **Klik op de dash in Ubuntu Desktop**  **zoek op "xterm" en open deze**  **vim oef5.sh**  **i**  **#!/bin/bash --**  **for ((i = 10; i >=1; i--))**  **do**  **echo -n "$i "**  **sleep 1**  **done**  **echo -e "\033[1;31;5;7mKAPOW\033[0m"**  **ESC**  **:wq**  **chmod u+x oef5.sh**  **./oef5sh** | OSX:Users:Fraa:Desktop:Screen Shot 2014-11-08 at 19.22.35.png |

● Schrijf een script oef6.sh dat alle directories die in de PATH variabele zitten afdrukt onder elkaar. *Tip: IFS gebruiken !*

student@desktop:~/bin$ ./oef6.sh

Overzicht directories in PATH

/usr/local/sbin

/usr/local/bin

/usr/sbin

/usr/bin

/sbin /bin

/usr/games

/usr/local/games

**Oplossing 1:**

**#!/bin/bash**

**echo Overzicht directories in PATH**

**IFS=:**

**for var in $PATH**

**do**

**echo $var**

**done**

**Oplossing2:**

**#!/bin/bash --**

**echo "Overzicht directories in PATH: "**

**echo $PATH | tr ':' "\\n"**

● Schrijf een script oef7.sh. Pas de vorige oefening aan, zodat per directory de laatste 5 entries (in deze directory) worden afgedrukt.

student@desktop:~/bin$ ./oef7.sh Overzicht directories in PATH Directory: /usr/local/sbin

Directory: /usr/local/bin

Directory: /usr/sbin vmware­checkvm vmware­rpctool vmware­vmblock­fuse vpddecode zic

Directory: /usr/bin zipnote zipspli zjsdecode zlib­flate zsoelim

Directory: /sbin wipefs wpa\_action wpa\_cli wpa\_supplicant xtables­multi

Directory: /bin zforce zgrep zless zmore znew

Directory: /usr/games espdiff gnome­mahjongg gnome­mines gnome­sudoku sol

Directory: /usr/local/games

**Oplossing 1:**

**#!/bin/bash --**

**echo "Overzicht directories in PATH: "**

**aantal=$(echo $PATH | tr ':' "\\n" | wc-l)**

**for ((i = 1; i <= $aantal; i++))**

**do**

**pad=$(echo $PATH | tr ':' "\\n" | head=$i | tail -1)**

**echo -ne "Directory $pad\t"**

**ls -Sr $pad | tail -5 | tr '\n' ' ' | echo ""**

**done**

**Oplossing 2:**

**!/bin/bash**

**echo "Overzicht directories in PATH: "**

**IFS=:**

**for var in $PATH**

**do**

**echo $var**

**echo $(ls $var | tail -5)**

**echo -e '\n'**

**done**● Schrijf een script oef8.sh dat het aantal .conf bestanden telt die in de directory /etc (recursief) voorkomen. Er wordt ook een bestand allconfsfrometcgemaakt en weggeschreven in de homedirectory van de user die de script uitvoert. Hierin staan de namen van al deze .conf bestanden, zonder dubbels. (Dubbels kunnen voorkomen, aangezien in subdirectories bestanden met dezelfde naam kunnen voorkomen.) Het aantal .conf bestanden met een unieke naam wordt ook geteld en afgedrukt.

student@desktop:~/bin$ ./oef8.sh

Aantal .conf files in /etc: 526

Aantal .conf files in /etc met unieke naam: 449

student@desktop:~$ head ­10 allconfsfrometc 10­antialias.conf 10­autohint.conf 10­console­messages.conf 10gsfonts.conf 10­hinting.conf 10­hinting­full.conf 10­hinting­medium.conf 10­hinting­slight.conf 10­ipv6­privacy.conf 10­kernel­hardening.conf

*Tip: gebruik eventueel ook een tijdelijk bestand, dat je wist op het einde van je script*. *Het laatste veld opvragen via cut lukt niet, wel het eerste. Gebruik daarom mogelijk het commando rev in combinatie van cut. (Zie man rev)*

**#!/bin/bash --**

**sudo ls -R /etc | grep -E '\*.conf' | uniq > /home/$USER/allconfsfrometc**

**echo "Aantal .conf files in /etc: $(sudo ls -R /etc | grep -E '\*.conf' | wc -l)"**

**echo "Aantal unieke .conf files in /etc: $(cat /home/$USER/allconfsfrometc | wc -l)"**

●  Maak een nieuw script aan, genaamd “overzicht\_homefolder”  Zorg er voor dat dit script altijd wordt uitgevoerd in de bash-shell, dat je auteurs-gegevens er in staan en dat het script de volgende uitvoer geeft: config  map  verborgen .vimrc  bestand  verborgen mijndir  map test  bestand uitvoerbaar.

**#!/bin/bash --**

**# Auteur: Voornaam Naam**

**# ls /home/$USER/ | tr '\n' ' '**

* ●  Geef uit je homefolder alle bestanden die eindigen op “.sh”  uitvoerbare rechten. Gebruik hiervoor de for-lus in een scriptje, genaamd “enable\_run\_for\_scripts.sh”. Zorg er voor dat het ook werkt indien er spaties in de bestandsnamen staan. TIP: zoek in manpage van bash naar IFS.

**#!/bin/bash --**

**aantal=$(ls ~/\*.sh | wc -l)**

**for ((i = 1; i <= $aantal; i++))**

**do**

**file=$(ls ~/\*.sh | head -$i | tail -1)**

**chmod u+x $file**

**done**

* ●  Maak voor iedere gebruiker die kan inloggen een directory met zijn loginnaam onder een (nieuwe) directory /tmp/backups/

**#!/bin/bash --**

**aantal=$(cat /etc/passwd | grep -v 'nologin' | cut -d':' -f1 | wc -l)**

**for ((i = 1; i <= $aantal; i++))**

**do**

**userfolder=$(cat /etc/passwd | grep -v 'nologin' | cut -d':' -f1 | head -$i | tail -1)**

**mkdir -p /tmp/backups/$userfolder**

**done**

* ●  Schrijf een script “userlist” dat een alfabetisch gesorteerde lijst met gebruikers toont.

**#!/bin/bash --**

**echo -e "Alle users: \n"**

**echo -e "$(cat /etc/passwd | cut -d':" -f1 | sort)\n"**

* ●  Schrijf een script “kopieer”. Dit script vraagt eerst achter de naam van  het te kopieren bestand. Vervolgens wordt er gevraagd waarnaar het bestand moet gekopieerd worden. Voer voldoende controles uit (bvb. bestaat het ingegeven bestand ?, ... ?)

**#!/bin/bash --**

**echo -n "Geef de naam van het bestand: "**

**read naam**

**echo -n "Geef de nieuwe locatie van het bestand op: "**

**read locatie**

**if [ -f $naam ]**

**then**

**cp $naam $locatie/$naam**

**else**

**echo "Het bestand bestaat niet!"**

**fi**

# Hoofdstuk 23 en 24

* ●  Pas je prompt aan, zodanig dat deze enkel uit het “>”-teken bestaat
* **PS1=">"**
* ●  Zorg ervoor dat de prompt opnieuw op de originele waarde komt, zonder uit te loggen, te herstarten of een sub-shell te starten. Hint: de prompt wordt gezet in ~/.bashrc
* **source ~/.bashrc**
* ●  Maak een script dat de naam van het script teruggeeft, alsook de eerste parameter: Let op: De naam van het script en niet het volledig pad. Voorbeeld:  student@desktop:~$ /home/student/bin/oef23\_03.sh 12345 De naam van het script is: Oef23\_03.sh De eerste parameter is: 12345

**#!/bin/bash --**

**echo "$(echo $0 | rev | cut -d '.' -f1,2 | rev | tr-d '/') De eerste parameters is: $1"**

* ●  Maak een script, genaamd grootstegetal.sh, dat het grootste getal teruggeeft van de getallen die als parameter worden meegegeven.

**#!/bin/bash**

**if [ $1 -gt $2 ]**

**then**

**echo "$1 is het grootste getal"**

**else**

**echo  "$2 is het grootste getal"**

**fi**

* ●  Pas het voorgaand script aan. Voeg een functie, genaamd syntax, toe. Deze functie toont een foutmelding en de juiste syntax waarmee het script moet worden aangeroepen  U hebt een onjuist commando opgegeven  Syntax: script.sh getal1 getal2 getal3 ... getaln De functie eindigt ook de uitvoer van het script met een foutcode.  Indien er geen parameter wordt meegegeven, moet de foutmelding en juiste syntax worden getoond.

**#!/bin/bash**

**check() {**

**if [ "$#" -eq "0" ]**

**then**

**echo "Foute syntax!  Juist: commando getal1 getal2" ; exit 1**

**else**

**grootst $1 $2**

**fi**

**}**

**grootst() {**

**if [ $1 -gt $2 ]**

**then**

**echo "#$1 is het grootste getal"**

**else**

**echo "$2 is het grootste getal"**

**fi**

**}**

●  Pas het voorgaand script aan. Controleer of alle parameters ook getallen zijn. Indien het niet allemaal getallen zijn, dan dient de juiste syntax getoond te worden.

* #!/bin/bash --

**if [ "$1" -eq "0" ]**

**then**

**echo "Foute syntax! Juist: commando getal1 getal2" ; exit 1**

**else**

**if [[ $1 =~ ^[[:digit:]]+$ ]]**

**then**

**if [[ $2 =~ ^[[:digit:]]+$ ]]**

**then**

**grootst $1 $2**

**else**

**echo "Foute syntax! Juist: commando getal1 getal2" ; exit 1**

**fi**

**else**

**echo "Foute syntax! Juist: commando getal1 getal2" ; exit 1**

**fi**

**fi**

**}**

**grootst() {**

**if [ $1 -gt $2 ]**

**then**

**echo "$1 is het grootste getal"**

**else**

**echo "$2 is het grootste getal"**

**fi**

**}**

**check $1 $2**

* ●  Maak een scriptje dat twee getallen als parameters ontvangt. In het script worden deze parameters doorgegeven aan een funtie, genaamd vermenigvuldig. Deze functie vermenigvuldigd de twee parameter-waarden met behulp van het commando “let” en stopt de uitkomst in een nieuwe variabele uitkomst. Print buiten de functie de waarde van de variabele uitkomst af naar het scherm.

**#!/bin/bash --**

**vermenigvuldig(){**

**let "uitkomst = $1 \* $2"**

**}**

**vermenigvuldig $1 $2**

**echo $uitkomst**

* ●  Probeer vorig scriptje, maar nu met volgende regel toe te voegen als nieuwe eerste regel van de functie: local uitkomst=0. Werkt dit nog? Waarom/Waarom niet?
* **Het script werkt niet meer omdat je met local uitkomst=0 een lokale variabele binnen de functie declareert. Hierdoor is de variabele niet meer bekend buiten de functie, waardoor de echo niet meer werkt.**
* ●  Los vorig probleem op door in de functie de waarde te tonen via het echo-commando. Vang deze echo op in een variabele buiten de functie (door gebruik te maken van een subshell) en print vervolgens deze variabele af buiten de functie.

**#!/bin/bash --**

**vermenigvuldig(){**

**local uitkomst=0**

**let "uitkomst = $1 \* $2"**

**product=$uitkomst**

**}**

**vermenigvuldig $1 $2**

**echo $product**

* ●  Maak een scriptje, genaamd list.sh, dat :
* – de opties a,d en l opvangt
* – de parameter a laat alle files zien (ook de hidden files)
* – de parameter l laat de files in een lijst zien ipv in kolommen
* – de optie d ontvangt ook een parameter (de directory om te listen)
* – een nederlandstalige foutboodschap toont indien een foute optie wordt opgegeven
* – dezelfde nederlandstalige foutboodschap toont indien de optie h wordt meegegeven

– een gewone ls doet van de huidige directory indien geen opties werden meegegeven

– een ls uitvoert met de nodige meegegeven opties van de eventueel meegegeven directory

**#!/bin/bash --**

**while getopts ":aldh" option;**

**do**

**case $option in**

**a)**

**ls -a**

**;;**

**l)**

**ls -l**

**d)**

**ls $OPTARG**

**;;**

**h)**

**echo "Foute optie meegegeven!"**

**;;**

**\*)**

**echo "Foute optie meegegeven!"**

**;;**

**esac**

**done**

● Maak een quiz die een random getal van 0 tot 9 laat raden. Geef bij het juist raden weer in hoeveel pogingen het is geraden. Indien het getal geraden wordt in 4 pogingen of minder, dan dien je de quizzer ook te feliciteren.

Voor een random getal te berekenen gebruik je volgende regels: RANDOM=$(date +%s) #date uitgedrukt in aantal seconden sinds 1/1/1970. *randomgetal=$(( RANDOM % 10 )) #rest na /10 (modulo 10) = getal van 0-9*

**#!/bin/bash --**

**RANDOM=$(date +%s)**

**randomgetal=$(( RANDOM % 10))**

**teller=1**

**echo -n "Raadt een getal tussen 0-9: "**

**read antwoord**

**while ! [ $antwoord -eq $randomgetal ]**

**do**

**echo -n "Fout! Geef opnieuw een getal in: "**

**read antwoord**

**let teller++**

**done**

**if [ $teller -lt "4" ]**

**then**

**echo "Gefeliciteerd!"**

**else**

**echo "Gefeliciteerd! U heeft het getal in $teller pogingen geraden!"**

**fi**