Samenvatting Server OS Advanced – Linux

# Local User Management

## Theorie

Men kan aan user management doen door de GUI, de CLI of door de configuratiebestanden handmatig aan te passen.

/etc/passwd bevat 7 velden met informatie:

1. Username
2. X (wachtwoord)
3. User id
4. Primary group id
5. Description
6. Home directory
7. Shell

Root is de superuser en heeft als id 0. /etc/default/useradd bevat de standaardwaarden voor een gebruiker. /etc/shadow bevat de wachtwoorden van alle gebruikers, dit is read-only voor de root en heeft 9 velden:

1. Username
2. Versleuteld wachtwoord
3. De dag waarom het wachtwoord het laatst gewijzigd was
4. Aantal dagen dat het wachtwoord nog niet veranderd moet worden
5. Wachtwoord vervaldatum
6. Aantal dagen op voorhand van de vervaldatum een melding moet gegeven worden
7. *~~Aantal dagen na de vervaldatum voor het account uitgeschakeld is~~*
8. *~~De dag dat het account uitgeschakeld was~~*
9. *~~Veld zonder enige nuttige betekenis~~*

Een wachtwoord is uitgeschakeld wanneer er een ! voor het wachtwoord in het /etc/shadow bestand staat. De inhoud van /etc/skel wordt gekopieerd naar elke nieuwe home.

/etc/group bevat alle groupen en heeft 4 velden:

1. Groupname
2. Versleuteld wachtwoord
3. Group id
4. Lijst met leden

|  |  |
| --- | --- |
| Eerste karakter (ls -l) | Type |
| - | Regular file |
| d | Firectory |
| l | Symbolic link |
| p | Named pipe |
| b | Block device |
| c | Character device |
| s | Socket |

Een bestand is standaard nooit uitvoerbaar. Note: zoek uit hoe een umask werkt…

Een sticky bit op een map is er om te voorkomen dat gebruikers bestanden wissen waarvan ze geen eigenaar zijn:

* t -> sticky bit + x
* T -> sticky bit, geen x voor anderen

Een setgid bit op een map is er om te verzekeren dat alle bestanden in de map dezelfde groepseigenaar hebben:

* s -> setgid + x
* S -> setgid, geen x voor groepseigenaar

## Commando’s

whoami toont je username

who toont je informatie over wie ingelogd is

who am i toont je informatie over wie ingelogd is in je huidige sessie

w toont wie ingelogd is en wat ze aan het doen zijn

id toont je user id, group id en een lijst van groepen waar je lid van bent

useradd een gebruiker toevoegen

useradd -m een gebruiker toevoegen en zijn home mee aanmaken

userdel een gebruiker verwijderen

userdel -r een gebruiker en zijn home verwijderen

usermod een gebruiker wijzigen

passwd een gebruiker een wachtwoord toewijzen

chpasswd een gebruiker een wachtwoord geven (echo naam:ww | sudo chpasswd)

openssl passwd een geëncrypteerd wachtwoord aanmaken

chage de standaardwaarden voor wachtwoord policies

usermod -L # de gebruiker # zijn account uitschakelen

usermod -s een gebruiker zijn shell wijzigen

chsh een andere manier om een gebruiker zijn shell te wijzigen

su verander van gebruiker

groupadd een groep aanmaken

groupmod een groep wijzigen

groupdel een groep verwijderen

groups toont een lijst van groepen waar de huidige gebruiker toe behoord

gpasswd geeft de controle van de groep aan een andere gebruiker

adduser wachtwoord kan onmiddellijk meegegeven worden + home auto aangemaakt

addgroup groepid wordt getoond na het uitvoeren van commando

chgrp wijzigt de groepseigenaar

chown wijzigt de gebruikerseigenaar

umask wijzigt de standaard permissies voor een bestand of map

# Inodes & File Links

## Theorie

Een gewoon bestand bestaat uit bytes. Tekstbestanden, data bestanden, uitvoerbare programma’s en gedeelde bibliotheken vallen hieronder. Een map bevat referenties naar andere bestanden. Character en block device bestanden laten programma’s met de hardware. Local domain sockets zorgen voor de connecties tussen processen (propere communicatie). Named pipes zorgen ook voor de communicatie tussen processen draaiend op dezelfde host (FIFO bestanden). Symbolic links zijn soft links.

De inode tabel bevat alle inodes, gemaakt bij creatie van het bestandssysteem. Elke inode heeft een uniek nummer.

De permissies van een symbolic link hebben geen betekenis. Als je de permissies van een symbolic link aanpast, pas je eigenlijk de permissies van het origineel bestand aan. De permissies van de symbolic link blijven ongewijzigd.

Symbolic links kunnen verwijzen naar eender waar, zelfs over een netwerk. Hardlinks kunnen alleen voorkomen op eenzelfde partitie.

## Commando’s

df -i toont de inodes op het bestandssysteem  
ln # ## maakt een hardlink naar # met als naam ##  
ln -s # ## maakt een softlink (symbolic) naar # met als naam ##  
find / -inum # 2>/dev/null zoekt hards link met inode #  
rm # verwijdert de link #

# Network Management

## Theorie

Netplan is een configuratie die zich in een yaml-bestand bevind. Dit vervangt de vroegere ifup/ifdown. Een server kan beter een statisch IP hebben. Door netplan apply te doen, maakt Ubuntu 2 bestanden aan: /run/systemd/network/<nr>-netplan-<nic>.network en /etc/resolv.conf.

Vroeger werden netwerkkaarten genoemd naar eth0, eth1, eth2 etc. Dit kon ervoor zorgen dat 2 netwerkkaarten wisselden na een reboot. Nu krijgt iedere netwerkkaart een eenduidige naam die bij elke reboot hetzelfde is (vb. ens33).

Teruggaan naar de oude namen (eth0, eth1 etc.) kan zo:

* sudo nano /etc/default/grub
* GRUB\_CMDLINE\_LINUX=”net.ifnames=0 biosdevname=0”
* sudo grub-mkconfig -o /boot/grub/grub.cfg
* sudo nano /etc/network/interfaces
* auto eth0  
  iface eth0 inet static
* sudo reboot

## Commando’s

sudo netplan apply wijzigingen in /etc/netplan/\*.yaml doorvoeren  
sudo systemctl restart NetworkManager herstarten netwerkservice (Desktop)  
sudo systemctl restart systemd-networkd herstarten netwerkservice (Server)  
networkctl huidige netwerkinstellingen bekijken  
ifconfig (deprecated) informatie over het netwerk opvragen en wijzigen  
ip informatie over het netwerk opvragen en wijzigen  
ip a s # informatie opvragen over nic #  
ip a s up informatie opvragen over nic’s die UP zijn (niet down)  
ip link set # down de nic # neerhalen  
ip link set # up de nic # terug online brengen (leest de netplan-yaml uit)  
ip addr del IP/24 dev ens33 één IP van de nic verwijderen  
ip addr flush dev ens33 alle IP’s van de nic verwijderen  
ip addr add IP/24 dev ens33 één ip bij de nic toevoegen  
sudo hostname # zet een nieuwe hostname namelijk # (tijdelijk)  
sudo hostnamectl set-hostname # zet een nieuwe hostname namelijk # (permanent)  
nano /etc/hosts hosts-bestand bewerken  
arp -a toont alle entries van de netwerkkaarten  
sudo arp -d IP verwijdert het IP uit de entries  
ip neighbor (ip n) toont alle buren van de netwerkkaarten  
ip n del IP dev ens33 verwijdert een IP van de ens33-kaart  
ip route (ip r) toont alle routes  
route (deprecated) toont alle routes  
netstat -r (deprecated) equivalent aan route  
sudo route add default gw IP voegt tijdelijk een route toe  
ping IP testen of een TCP/IP verbinding toegankelijk is

# Process Management

## Theorie

Een process is een programma dat een bepaalde job uitvoert. Een PID is het id van een proces en is uniek (0 – 65535). Een PPID is het id van de parent van een process. Een child process wordt gestart door zijn parent. systemd wordt gestart door de kernel en heeft geen parent, zijn id is 1 (soms aangeduid met init). Om een process te stoppen gebruik je kill. Een deamon is een process dat bij het opstarten van het systeem start en continu blijft draaien. Een zombie is een afgesloten process dat nog steeds zichtbaar is.

$$ is een omgevingsvariabele die het huidige process id terug geeft. $PPID geeft de parent id terug.

Als je een process forkt, start dezelfde parent een process identiek aan het process dat je forkt. Het enige verschil is de id. Een exec vervangt het programma dat het huidige process uitvoert. Er wordt geen nieuw process gestart meer het huidige process start een ander programma.

Ieder process heeft een priority en een nice waarde. Een hogere nice waarde betekent een hogere priority waarde en een hogere priority waarde betekend minder CPU tijd. Normale gebruikers kunnen nice waarden toekennen van 0-20. Root kan toekennen tot -20 (opgepast!). Verlagen van een nice waarde moet ook altijd als root gebeuren.

Het plusteken bij jobs betekent dat dat het laatste en default process is. Een minteken betekent het voorlaatste process (wordt plusteken als default process eindigt).

## Commando’s

pidof # geeft alle processenid’s terug met de naam #  
ps geeft een lijst van de huidige processen  
pgrep # geeft het processid terug waarbij # de naam is  
top geeft een geordende lijst van de huidige processen  
kill een process afsluiten  
kill -l een lijst van mogelijke kill commando’s weergeven

|  |  |  |  |
| --- | --- | --- | --- |
| Commando | Cijfer | Optie | Betekenis |
| kill | -1 | Sighup | Process laten weten dat het de configuratiefile moet herlezen |
| kill | -15 | Sigterm | Vriendelijke vragen om af te sluiten (standaard optie) |
| kill | -9 | Sigkill | Kernel vragen het process te stoppen (kernel doet bam weg) |
| kill | -19 | Sigstop | Process pauzeren (geen CPU-cycles, wel nog geheugen) (vb. Ctrl + Z) |
| kill | -18 | Sigcont | Een gepauzeerd proces terug starten (reanimeren) |
| kill | -2 | Sigint | Process beëindigen of afbreken = interrupt (vb. Ctrl + C) |
| pkill |  |  | Process afsluiten met pattern matching |
| killall |  | Sigterm | Alle processen met de opgegeven naam beëindigen |

nice -n # ## geeft bij het starten van process ##, direct een nice waarde van #  
renice +# ## process met id ## en nieuwe nice waarde namelijk # geven.  
jobs toont de process die in de background van *JE HUIDIGE SHELL* draaien  
& process gevolgd door een & worden in de achtergrond uitgevoerd  
fg # background process naar foreground brengen, # is nummer backg job  
bg # gepauzeerd background process terug starten, # is nummer backg job

# Disk Management

## Theorie

Een harde schijf is een block device. Gegevens worden uitgelezen en opgeslagen per blok. Een systeem kan maximum 4 primaire partities hebben. Hiervan kan er eentje extended zijn en die partitie kan oneindig veel logische partities hebben. Een UUID of Universally Unique Identifier is een 128bit unieke id om objecten aan te duiden.

Stappenplan partities maken:

1. Herkennen van de harde schijven (sudo fdisk -l /dev/sd\*)
2. HD openen met fdisk (sudo fdisk /dev/sdb)
3. Bekijken van de huidige partitietabel (p)
4. Bekijken van de vrije ruimte (F)
5. Toevoegen van partities (n -> p primary of e extended -> 1 -> enter)
6. Bekijken van de huidige partitietabel (p)
7. Eventueel type van partitie wijzigen (t)
8. Opslaan van de nieuwe partitietabel (w)

De MBR (Master Boot Record) bevant de partitietabel. De huidige versie van ext is ext4.

In /proc/filesystems kan je een lijst met toekenbare bestandssystemen vinden. (nodev: not mounted on a block device) In /proc/mounts kan je een lijst met alle mounts vinden. In /proc/partitions staat info van de kernel in bestanden en mappen. Hier staat ook info over de partities die door de kernel gekend zijn. In /etc/mtab staand de mounts onderhouden door mount zelf. In /etc/fstab staan alle mounts die automatisch bij een boot gemount moeten worden.

Standaard wordt 5% van een filesysteem gereserveerd voor bestanden van root en daemons die als root draaien. Voor heel grote filesystemen is 5% echter te veel. We kunnen dit aanpassen met tune2fs.

## Commando’s

lsblk toont een lijst van alle block devices (list block)  
blkid toont de UUID van alle block devices (block id)  
ls -l toont de UUID van alle block devices  
file -s toont de UUID van alle block devices  
fdisk -l toont een lijst van alle HDs en hun partities  
fdisk -l /dev/sda toont een overzicht van de configuraties van sda  
dmesg toont een lijst van alle kernel boot messages (incl. detectie HDs)  
lshw toont een lijst van alle hardware (list hardware)  
lsscsi toont een lijst van alle SCSI-apparaten (list SCSI)  
parted tool om met partities te werken  
gparted grafische tool om met partities te werken  
sfdisk -d /dev/sdx > partitietabel.sdx.sfdisk backup van de partitietabel maken  
sfdisk /dev/sdx < partitietabel.sdx.sfdisk terugzetten van de partitietabel  
mkfs.ext4 /dev/sdb1 bestandssysteem ext4 op een partitie sdb zetten  
tune2fs parameters van ext2, ext3 en ext4 zien en/of aanpassen  
fsck een bestandssysteem controleren op fouten  
mount gemountte bestandssystemen tonen  
 ro mounten als read only  
 noexec op de mount kunnen geen scripts uitgevoerd worden  
 nosuid er wordt geen rekening gehouden met setuid-bits  
 noacl er kunnen geen acl-rechten gelegd worden  
mount # ## een bestandssysteem # linken aan een map ##  
umount # een bestandssysteem ontlinken, # kan het bestsyst en de map zijn  
df geeft de vrije ruimte weer (diskspace free)  
du geeft de groottes van mappen en partities weer (disk usage)  
badblocks data veilig overschrijven (4 keer)  
dd if=# of=## bestanden kopiëren en converteren of overschrijven met nullen  
nwipe data veilig van een ssd overschrijven

# RAID

## Theorie

RAID of Redundant Array of Independent Disks is het verenigen van meerdere HDs of partities voor snelheidswinst of veiligheid van de data te vergroten. Dit kan zowel softwarematig als hardwarematig zijn. De verschillende RAID-levels geven het type RAID weer (vb. RAID0).

RAID0: Data wordt parallel naar 2 of meerdere disks geschreven. De data wordt verdeeld (striping). Het voordeel is dat je snelheidswinst hebt en alle ruimte beschikbaar is. Het nadeel is dat als 1 disk crasht, alle data verloren is.

RAID1: Data wordt per 2 disks gespiegeld (mirroring). Het voordeel is dat er een grotere betrouwbaarheid is, daarnaast is het eenvoudig en goedkoop. Het nadeel is dat je maar de helft van de capaciteit kan gebruiken.

RAID5: Data wordt in blokken weggeschreven over verschillende drives (min. 3). Op iedere RAID member wordt een parity block geplaatst van de datablocks op de andere RAID members. Dit is striping met roterende pariteit. Het voordeel is dat het betrouwbaar is een en een goede snelheid heeft. Het nadeel is dat het trager is dan mirroring.

RAID 6: Zelfde als RAID5 maar pariteit wordt op 2 schijven weggeschreven. Dus minimum 4 disks en dan mogen er ook 2 schijven falen.

RAID0/1: mirror van stripes. Eerst worden er 2 RAID0 stripes gemaakt en dan worden deze gemirrord. Meerdere failures zijn geen probleem, als er maar een gespiegelde tegenhanger clean blijft.

RAID1/0: stripe van mirrors. Eerst worderen de mirrors gemaakt en dan worden deze gestriped. Er mogen tot 3 disks failen zolang deze maar niet in eenzelfde mirror zitten.

RAID5/0: stripe van RAID5-arrays. Eerst worderen de RAID5-arrays gemaakt en dan worden deze gestriped.

Een type goedzetten in de partitie gebeurd met fdisk. Fdisk -> t -> 1 -> fd (RAID autodetect) -> w. Een RAID-set wordt soms na rebooten automatisch hernoemd naar md127. We kunnen dit tegengaan door:

1. sudo su
2. mdadm –detail –scan >> /etc/mdadm/mdadm.conf
3. -> ARRAY /dev/mdo UUID=…
4. update-initramfs -u
5. shutdown -r now

Een spare is een block-device dat gewoon wacht totdat een RAID-member failed. Dan wordt hij actief en zal hij de failed member vervangen. (Rebuilden) Deze spare moet ook een partitie RAID autodetect (fd) bevatten.

## Commando’s

sudo mdadm --examine /dev/sdc /dev/sdd Disks controleren  
sudo mdadm --examine /dev/sdc1 /dev/sdd1 Partities controleren  
sudo mdadm --create /dev/md0 --level=mirror RAID maken  
 --raid-devices=2 /dev/sdc1 /dev/sdd1  
sudo mdadm --detail /dev/md0 Details bekijken  
sudo mdadm –detail –scan Alle arrays bekijken  
sudo mdadm --manage /dev/md0 --add /dev/sde1 Spare toevoegen  
sudo mdadm --grow --raid-devices=3 /dev/md0 Spare een active member maken  
sudo mdadm --fail /dev/md0 /dev/sdd1 RAID member als failing aanduiden  
sudo mdadm --remove /dev/md0 /dev/sdd1 Gefaalt RAID member verwijderen  
sudo mdadm --grow --raid-devices=2 /dev/md0 Aantal members veranderen  
sudo mdadm --zero-superblock /dev/sdd1 Verwijderde member opn klaarmaken  
sudo mdadm --create /dev/md1 --level=5 RAID5 maken  
 --raid-devices=3 /dev/sdf1 /dev/sdg1 /dev/sdh1  
sudo mdadm --stop /dev/md1 volledige RAID stoppen

# LVM (Logical Volume Management)

## Theorie

Klassieke partities zijn niet flexibel genoeg om mee te werken. Als een partitie vol zit, zit je met de handen in het haar. LVM komt hier als een ridder op het witte paard tot de redding. Een volume group beheert physical volumes en logical volumes. Fysieke volumes zijn schijven, partities, RAID etc. Logische volumes zijn ‘block deviced’ met data. Men kan een LV heel snel en makkelijker verkleinen en vergroten.

Stappenplan:

1. Partitie aanmaken via fdisk
2. Apparaat toevoegen aan LVM (pvcreate)
3. Volume group aanmaken (vgcreate vg)
4. Een logisch volume maken (lvcreate –size # vg)
5. Bestandssysteem aanmaken (mkfs.ext4 /dev/vg/lvol0)
6. Map maken en mounten
7. Bestanden overzetten naar nieuwe mount

## Commando’s

sudo pvcreate /dev/sd\* apparaat toevoegen aan LVM (fysiek volume aanmaken)  
sudo lvmdiskscan [ | grep sd] lijst van block devices weergeven die met LVM kunnen werken  
sudo vgcreate vg /dev/sd\* aanmaken van een volume group  
sudo vgs alle volume groups weergeven  
sudo vgscan alle schijven scannen voor VG’s en /etc/lvm/.cache bijwerken  
sudo vgdisplay gedetailleerde informatie over een VG krijgen  
sudo pvs alle fysieke groepen weergeven  
sudo pvscan alle schijven scannen voor PV’s  
sudo pvdisplay gedetailleerde informatie over een PV krijgen  
sudo lvcreate –size # vg een logisch volume aanmaken met grootte # als bv. 500m, 5g …  
sudo lvs alle logische groepen weergeven  
sudo lvscan alle schijven scannen voor LV’s  
sudo lvdisplay gedetailleerde informatie over een LV krijgen  
sudo lvextend -L +100 /dev/vg/… 100mb toevoegen aan een LV (-L is idem aan --size)  
sudo lvrename vg/old vg/new Hernoemen van een LV  
sudo lvremove vg/new Verwijderen van een LV  
sudo pvremove /dev/sd\* Verwijderen van een PV  
sudo vgextend vg /dev/sd\* Extra PV toevoegen aan VG vg  
sudo vgreduce vg /dev/sd\* Verwijderen van een PV in een VG  
sudo vgchange # vg1 Een VG aanpassen zodat #:  
 -xy deze vergroot of verkleinbaar is  
 -xn deze niet vergroot of verkleinbaar is  
 -l 5 een maximum van 5 LV’s kan hebben  
 -p 5 een maximum van 5 PV’s kan hebben  
sudo vgremove vg de volume group vg verwijderen

# SSH & VNC

## Theorie

OpenSSH-client is standaard geïnstalleerd op Desktop en Server. OpenSSH-server is niet standaard geïnstalleerd behalve op de Server Cloud edition. De configuratie van SSH kunnen we vinden in: /etc/ssh/sshd\_config. Deze heeft als opties oa.

* ListenAddress Indien we op een bepaalde NIC willen luisteren
* MaxSessions Hoeveel gelijktijdige connecties toegelaten worden
* PermitRootLogin Op “no” voor security (na login sudo...)
* DenyUsers Deze gebruikers mogen niet inloggen over SSH
* DenyGroups De gebruikers van deze groepen mogen niet inloggen
* PasswordAuthentication Geeft aan of er met een passwoord ingelogd mag worden
* PrintMotd de motd in /etc/motd af laten drukken
* ServerAliveInterval zorgen dat de connectie behouden blijft

SSH gebruikt standaard poort 22. Authentication wordt uitgevoerd door een public key van de server te gebruiken om zich bij de client te authenticeren. Deze staat in /etc/ssh/ssh\_host\_#\_key.pub waarbij #, rsa, dsa, ecdsa of ed25519 kan zijn. De instelling StrictHostKeyChecking op de client staat standaard op Ask. Dit geeft bij de eerste verbinding de vraag of je wel naar die server wil connecteren.

Men kan ook passwordless inloggen. Dan wordt er een private/public-keypair gemaakt. De private key blijft op de client en is persoonlijk. De public key wordt naar de homedir van de user waarbij we willen inloggen gekopieerd. Een public key kan herbruikt worden dus moet men meestal maar 1 paar aanmaken.

De volgorde van gebruikers toegang is: DenyUsers, AllowUsers, DenyGroups, AllowGroups. Een extra beveiliging is in /etc/hosts.allow en /etc/hosts.deny (sshd: ALL). Indien het bestand /etc/nologin bestaat kan niemand inloggen behalve de root. De inhoud van dit bestand wordt ook getoond. Dit is handig wanneer er werken aan de server zijn en niemand mag inloggen.

Door gebruik te maken van SSH-X11 forwarding kunnen we grafische applicatie renderen op de server. Men kan ook aan Desktop Sharing doen via VNC (Virtual Network Computing)

## Commando’s

sudo apt install openssh-server OpenSSH-server installeren  
ss -l 'sport = ssh' toont dat er enkel geluisterd wordt via TCP op Port 22  
ss -lt4 toont listening Port ssh  
ss -lt4n toont listening Port 22  
ss -at4 toont zowel de listening, als de established  
ss -o state established '( dport = ssh or sport = ssh )' toont alle verbonden connecties van enkel Port 22  
ssh naam@serverip verbinden met een bepaalde user op een server  
ssh server verbinden op een server  
ssh-keygen -t rsa keypair aanmaken  
ssh-copy-id -i ~/.ssh/id\_rsa.pub user@ip public key kopiëren naar de server  
sudo systemctl reload ssh ssh deamon opnieuw opstarten  
sudo systemctl stop ssh ssh deamon stopzetten  
sudo /usr/sbin/sshd -ddd ssh in debugging modus opstarten  
ssh-agent bash nieuwe shell openen met de agent running  
ssh-add ~/.ssh/id\_rsa private key in het geheugen zetten  
ssh user@ip ‘commando’ commando uitvoeren op de server, daarna stopt con  
ssh -t user@ip ‘commando’ commando uitvoeren als er een tty nodig is  
scp lokaalBestand user@ip:doelmap bestand kopiëren naar de server (of andersom)  
sftp user@ip bestande kopiëren over sftp  
sshfs user@ip:sshmount/ map over ssh lokaal mounten  
sudo fusermount -u sshmount sshfs-mount unmounten

# Scripting

## Theorie

Een interpreter geeft aan wat er gebruikt moet worden om de onderstaande commando’s te begrijpen. Dit wordt ook wel een she-bang genoemd (#!/bin/bash), je kan - - erachter plaatsen zodat er geen andere commando’s meer uitgevoerd kunnen worden (veiligheid).

Operatoren test-commando:

* -lt less than
* -gt greater than
* -ge greather or equal to
* -le less or equal to
* -eq equal to
* -ne not equal to
* = equals a string
* != Not equals a string
* -d does directory exist
* -f does file exist
* -a AND-operator
* -o OR-operator

If-structuur:

if  
 then  
 else  
 fi

if  
 then  
 elif  
then  
else  
then  
 fi

For-lus:

for  
 do  
 done

Deze kan zijn voor losse items: for teller in 1 2 3 4 5  
Deze kan zijn voor een range: for teller in {1..5} of for teller in `seq 1 5`  
Deze kan zijn voor bestanden door file-globbing: for file in `ls \*`

While-lus:

while  
 do  
 done

Until-lus:

until  
 do  
 done

Standaard wordty een script in een subshell uitgevoerd. Dit zorgt ervoor dat je variabelen niet meer bestaan als je script gedaan is. Je kan een script uitvoeren in de huidige shell (sourcing a script). Dit kan je aanroepen door ‘source scriptnaam.sh’ in te geven. Hiervoor heb je geen uitvoerrechten nodig dan!!

Je kan parameters meegeven met een script. De verwijzing ernaar is $1, $2, $3 etc. met een maximum van 9 parameters. $10 aanroepen geeft geen foutmelding maar garbage.

Een shell functie is een groep van commando’s dat aangeroepen kan worden door de functienaam. Dit zorgt voor een centraal onderhoud, gaat herhaling van code tegen en zorgt voor snelheid (uit werkgeheugen ipv schijfoperatie). Functies moeten altijd bovenaan het script staan. Functies werken ook met parameters. Waneer je in een functie voor een variabele ‘local’ zet, wordt deze niet in de main aangepast.

function naam naam ()  
{ OF {  
 // commando // commando  
} }

## Commando’s

# commentaar geven  
chmod u+x scriptnaam maakt het script uitvoerbaar  
echo -n “Tekst” print tekst af en laat de cursor op dezelfde lijn staan  
read vGetal vraagt input van de gebruiker en slaagt dit op  
echo “$vGetal” print de waarde van de variabele af  
test $vGetal -gt 100 && echo “Groter” waarden vergelijken  
 || echo “Kleiner” [ $vGetal -gt 100 ] is de verkorte versie  
  
$0 commandonaam  
$# geeft het aantal gegeven parameters  
$\* geeft alle parameters gescheiden door de systeem-IFS  
$@ geeft alle parameters in tabelvorm (elke parm is string)  
$? geeft de laatste return-code  
$$ geeft het proces-ID van het script  
shift schuift alle parameters een positie naar links ($0 blijft hetzelfde en $1 gaat verloren)  
=~ duidt op regex-expressie  
[[ … ]] duidt op regex-expressie  
exit # beëindig het programma met code #  
return # beëindig de functie met argument #  
getopts “:ab:c” option geeft alle opties en steekt in var option  
$OPTARG geeft argument van optie  
$OPTIND geeft het volgnummer van de volgende optie of arg  
$((…)) wiskundige bewerking doen en resultaat gebruiken  
eval argumenten van eval worden tot 1 string samengezet  
(( … )) wiskundige testen  
let voert rekenkundige berekeningen uit

# AWK

## Theorie

AWK is een scripttaal die bedoeld is voor het automatisch verwerken van tekst. AWK is lijn georienteerd, dit betekent dat elke regel gebruikt wordt als input. Een patroon wordt getest op elke regel, indien een match, wordt de actie op die regel uitgevoerd, indien geen patroon is opgegeven, wordt elke regel geselecteerd.

awk '  
BEGIN { actions }  
/pattern/ { actions }  
/pattern/ { actions }  
END { actions }  
' file(s)

BEGIN wordt aan het begin voor het uitlezen van de regels uitgevoerd. END wordt aan het einde na het uitlezen van de regels uitgevoerd.

{ print $0 } of { print } print de hele regel af.  
{ print $1 $2 $3 } print zonder spaties af.  
{ print $1,$2,$3 } print met spaties af.

AWK variabelen:

NR lijnummer  
NF aantal velden  
FILENAME naam van het bestand  
FS field seperator  
RS record seperator  
OFS output field seperator  
ORS output record seperator

Stringbewerkingen:

length(x) geeft de lengte van x terug of indien geen x is meegegeven, de lengte van de lijn  
substr(s, pos, c) geeft pos met lengte c in s terug  
index(s1, s2) geeft de positie van s2 in s1, indien niet gevonden geeft het 0 terug  
split(s, a) split s in een array a en geeft het aantal velden terug (FS is de field seperator)  
system(“cmd”) voert het UNIX-commando uit en geeft zijn exit status terug  
  
toupper(s) naar hoofdletters  
tolower(s) naar kleine letters  
delete array[e] verwijdert het element e in de array  
sub(r, s [,t]) vervangt r door s in t  
gsub(r, s [,t]) sub-commando maar dan globaal ipv 1 keer  
match(s, r) zoekt in s naar r, geeft de index hiervan terug

If, For, While, Do While en Functie- constructies zijn gelijkaarding in syntax aan Java.

## Commando’s

awk -F: ‘…’ voert AWK uit met field seperator ‘:’ en commando ‘…’  
awk -f file voert de AWK-code uit die in je file staat  
x ~ y x gelijk aan regex y  
getline vIngave < "/dev/tty" vraagt input

# Package Management

## Theorie

Software wordt aangeboden in pakketvorm. Een package met .deb of .rpm bevat de eigenlijke software maar ook metadata zoals de naam, versienr, dependencies etc. Een software repository is een lijst met beschikbare software voor een bepaalde distributie. Het bevat ook een locatie waar de software packages afgehaald kunnen worden.

DPKG staat voor Debian Package Management. Dit zorgt voor de installatie van .deb bestanden en beheert geïnstalleerde packages. APT staat voor Advanced Packaging Tool. Dit is een front-end tool voor DPKG. Er bestaat ook een GUI front-end voor apt, dit kan Ubuntu Software Center of Synaptic zijn. RedHat werkt met .rpm packages die op yum draaien.

PPA staat voor Personal Package Archives en zijn repositories die standaard niet in Ubuntu zitten.

Installeren van source:

1. Downloaden en uitpakken van source-code
2. ./configure (checken van systeemvariabelen, nodige libraries etc.)
3. make (compileren van de source-code mbv de Make-file)
4. sudo make install (gecompileerde bestanden in de juiste mappen kopiëren)

Een snap is een bundel van een app met zijn dependencies in een eigen filesysteem (Ubuntu Core). Dit werkt zonder aanpassingen op meerdere distro’s. Het draait in zijn eigen bubbel en is afgescheiden van andere snaps. snapd is de deamon die zorgt voor de juiste werking van snaps. Een snapbestand eindigt op .snap en heeft een gecompresseerd bestandssysteem (squashfs). Dit wordt dynamisch gemount via loopback devices. Snaps worden geïnstalleerd in /snap/bin.

## Commando’s

dpkg -l # lijst alle packages op met naam # (regex mag)  
dpkg -l toont alle geïnstalleerde packages  
dpkg -p # toont informatie van de package #  
dpkg -S # zoekt welke package het bestand # heeft geïnstalleerd  
dpkg- L # toont welke bestanden geïnstalleerd zijn door programma #  
wget -q url.deb .deb bestand binnenhalen (downloaden)  
dpkg -i # bestand # (.deb) installeren  
dpkg -r # programma # verwijderen  
dpkg -P # programma # en config-bestanden verwijderen  
apt help toont beknopte help over apt  
apt update zoekt naar wijzigen via /etc/apt/sources.list  
apt install # zoekt naar # in apt-cache (/var/lib/apt/lists/…) en installeerd  
apt clean maakt /var/cache/apt/archives leeg  
apt remove # verwijdert het programma #  
apt purge # verwijdert het programma # en de config-bestanden  
apt autoremove verwijdert dependencies die niet meer nodig zijn  
apt upgrade update alle software die geïnstalleerd is  
apt list toont alle packages die geïnstalleerd kunnen worden  
apt search # doorzoekt de APT-cache (geen internet nodig)  
apt show # toont informatie over package #  
apt edit-sources shortcut voor ‘nano /etc/apt/sources.list’  
apt-add-repository ppa:# voegt de PPA # toe  
apt-add-repository -r ppa:# verwijdert de PPA #  
tar -cf /tmp/home.tar /home maakt een tar-archief van home  
tar -czf /tmp/home.tar /home maakt een tarball (gzip) van home  
tar -cjf /tmp/home.tar /home maakt een tarball (bzip2) van home  
tar -xf home.tar pakt de tar uit in de huidige directory (-C dirNaarTeSchrijven)  
snap find # zoekt naar snaps met # in de naam  
snap list # [--all) lijst van geïnstalleerde snaps voor #  
snap list [--all] lijst van geïnstalleerde snaps  
snap info # geeft info over de snap van #  
snap install [--kanaal] # installeert snap van #, kanalen (stable, candidate, beta, edge)  
snap refresh # updaten van een snap (gebeurd standaard dagelijks)  
snap switch --channel=kanaal # wisselt het kanaal van # naar (stable, candidate, beta, edge)  
snap refresh --channel=kanaal # wisselt het kanaal van # naar (stable, candidate, beta, edge)  
snap run # starten van snap #  
snap revert # snap # terugdraaien naar een vorige versie  
snap enable # activeren van snap #  
snap disable # deactiveren van snap #  
snap remove # verwijderen van snap #  
snap changes geeft een lijst van recente wijzigingen

# Scheduling

## Theorie

In /etc/at.allow staat een lijst van gebruikers die jobs mogen plannen. Wie hier niet in staat, mag er geen plannen. Een lege at.allow wil zeggen dat niemand jobs mag plannen. Als deze file niet bestaat wordt at.deny gecontroleerd.

In /etc/at.deny staat een lijst van gebruikers die geen jobs mogen plannen. Wie niet in deze file staat, mag jobs plannen. Een lege at.deny wil zeggen dat iedereen jobs mag plannen. Als beide files niet bestaan, mag iedereen jobs plannen.

Je kan ook jobs herhalen op specifieke momenten, dit kan met crontab. Deze staan opgeslagen in /var/spool/cron/crontab/<username>. Je hebt 5 velden, om een tijdstip te bepalen (minute hour dayOfMonth month dayOfWeek). Een alternatief is @reboot, @yearly, @annually, @monthly, @daily, @midnight, @hourly.

Voor crontab bestaan er ook cron.allow en cron.deny bestanden. In /etc/crontab kan je de algemene crontab file vinden.

## Commando’s

at # zal uitgevoerd worden op tijdstip #  
at -l geeft een lijst van alle geplande jobs (zelfde als atq)  
atrm # verwijdert job met id # uit de wachtrij  
crontab -e bewerken van de crontab file  
crontab -l inhoud van de crontab file weergeven

# Memory management

## Theorie

Door ‘head -5 /proc/meminfo’ uit te voeren kunnen we het geheugen en de cache tonen. Dit geeft ons volgende velden:

* MemTotal Totale hoeveelheid fysieke RAM
* MemFree Ongebruikte RAM
* MemAvailable Hoeveelheid RAM om programma’s op te starten zonder SWAP
* Buffers RAM gebruikt voor buffering files
* Cached Hoeveelheid RAM gebruikt als cache

Swapfile aanmaken:

1. sudo dd if=/dev/zero of=/swapfile2 bs=1024 count=2048000
2. sudo mkswap /swapfile2
3. sudo chmod 0600 /swapfile2
4. sudo swapon /swapfile2

## Commando’s

free -h geheugen en cache weergeven  
top informatie over het geheugen weergeven  
cat /proc/swaps swap weergeven  
sudo mkswap /dev/sb1 swap aanmaken  
sudo swapon /dev/sb1 swap activeren  
sudo swapoff /dev/sb1 swap deactiveren

# Logging

## Theorie

De daemon voor systeemlogging is rsyslogd. Dit staat voor Rocket-Fast System for Log processing. De configuratie is terug te vinden in /etc/rsyslog.conf. Je kan zien wat standaard naar waar gelogd wordt in /etc/rsyslog.d/50-default.conf. De opbouw bestaat uit ‘faciliteit.prioriteit /pad’.

Belangrijke logfiles:

* /var/log/syslog Generale logging-berichten
* /var/log/auth.log User login en authorizatie’s
* /var/log/faillog Bevat foutieve login-pogingen
* /var/log/boot.log Logs van opstartscripts van vorige boots
* /var/log/kern.log Bevat kernel-logs
* /var/log/dmesg Dump van kernel-message-buffer
* /var/log/btmp Logs van mislukte loginpogingen
* /var/log/wtmp Logs van login, logout, reboot en shutdown
* /var/log/apport.log Logs van crashes van applicaties
* /var/log/dpkg.log Logs van Package Management Systeem
* /var/log/secure Logs van ssh, proftp etc.
* /var/log/<daemon>/ Logs van een bepaalde daemon

De logrotate-daemon zorgt voor rotatie, verwijdering en compressie van logbestanden. De algemene config is te vinden in /etc/logrotate.conf.

## Commando’s

rsyslogd -v versie van logging systeem weergeven  
faillog logfile van foutieve login-pogingen bekijken  
dmesg kernel-logs tijdens booten bekijken  
lastb mislukte login-pogingen tonen  
last Logs van login, logout, reboot en shutdown tonen  
lastlog laatsts login-tijd per gebruiker  
logger “tekst” tekst naar syslog sturen  
logger -t # “tekst” tekst naar syslog sturen en script of daemon # afzender maken