Linux Werkcollege 1

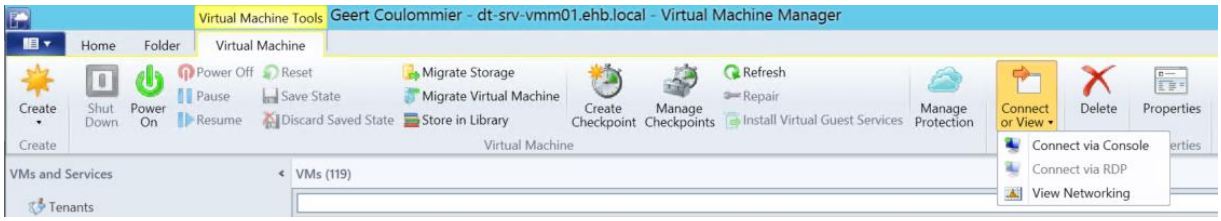
1. Doel van deze werkcolleges

In deze werkcolleges zullen we voornamelijk het besturingssysteem Linux nader bekijken. Enerzijds omdat dit een inleiding is voor het werken met Linux in de volgende jaren, anderzijds omdat bepaalde aspecten in verband met besturingssystemen in Linux duidelijker zijn dan in Windows. In Linux is het mogelijk rechtstreeks door te dringen tot het hart van het systeem, hetgeen veel leerrijker is dan enkel de vertrouwde grafische omgeving van Windows. Verder zijn de meeste webservers en mailservers op het internet Linux servers, dit door de eenvoud van configuratie en de stabiliteit van het systeem. Een kennis van Linux is dus onontbeerlijk voor elke IT-er.

Ook in Linux is het natuurlijk mogelijk via een grafische omgeving te werken. Dit gaan we nu echter niet doen, we duiken in deze werkcolleges als het ware achter de vensters.

Belangrijk: Mac OS X - Het besturingssysteem van Mac, is net als Linux gebaseerd op Unix. De meeste commando's die we in deze werkcolleges zien, kunnen we dus ook gebruiken in een Mac omgeving via de terminal.

Voor deze werkcolleges zullen we werken met individuele Linux virtuele machines. Om VM's aan en uit te zetten, maken we gebruik van de Virtual Machine Manager, een applicatie te vinden in het startmenu van onze EhB-windows computers. Eens deze is opgestart, kan je verbinding maken met de virtualisatiecluster (waar je virtuele machines opstaan) met je persoonlijke EhB-credentials.Hier zal je al je VM's zien staan, waaronder je Linux VM.. Klik op de groene 'Power On'-knop om je virtuele machine op te starten, en vervolgens op 'Connect or View'-knop met optie 'connect via console'. Een nieuw scherm gaat open. Dit is het scherm van je linux VM alsof je fysiek aan deze computer zou zitten.



Je kan aanloggen op de virtuele machine met de credentials zoals vermeld in het document op Canvas speciaal daarvoor aangemaakt onder "Introductie – labo setup".

Let op, in linux is zowat alles hoofdlettergevoelig: commando's, bestandsnamen, maar dus ook de username en het paswoord!

Er is echter een veel betere manier om deze machine vanop afstand te beheren, namelijk via het SSH-protocol.

Hiervoor heb moet op de machine die je vanop afstand wil beheren Open-SSH zijn geïnstalleerd (wat bijna altijd het geval is), en op de computer vanwaar je de andere computer wil beheren een SSH-client zoals Windows Terminal.

Opdracht:

1. Verbind met je Linuxmachine zoals aangegeven in de les, zorg ervoor dat je het SSH protocol hebt geselecteerd.
   De syntax voor ssh in een cli-omgeving is als volgt:
   ssh username@computername -p poort

   ssh: het commando
   username: de username op de remote computer
   computername: de naam of het ipadres van de remote computer
   poort: de poort langswaar je toegang krijgt op de remote computer. Aangezien we SSH gebruiken is deze normaal gezien 22, maar het kan ook anders zijn, afhankelijk van de situatie.
   Bijvoorbeeld: ssh student@10.3.50.34 -p 3300
2. Log in met dezelfde credentials als hierboven.
3. Maak een folder in de homedirectory met jouw voornaam als naam. Je kan dit doen met het commando:
   mkdir [vervang_dit_door_je_voornaam]
4. Maak nu ook verbinding met dezelfde linux-box via winscp.
5. Maak in notepad op je windows-computer een tekstbestand aan.
6. Kopieer dit bestand over naar je persoonlijke folder onder homefolder die je hebt aangemaakt in 3.
7. Verifieer via je ssh terminal of dit bestand goed is aangekomen.

2. De commandoshell Bash

Iedereen is nu ingelogd in het systeem.

Wat je nu ziet is een **command prompt** met daarachter een knipperende cursor. Je bent terecht gekomen in de **commandoshell** van Linux, bash. Deze commandoshell is goed te vergelijken met MS-DOS. Op Linux serversystemen wordt nog steeds het meest met de commandoshell gewerkt, enerzijds omdat men zo sneller kan werken dan met vensters en heel gemakkelijk commando's kan automatiseren met behulp van scripts, anderzijds omdat het systeem dan niet nodeloos belast wordt met een grafische omgeving die veel geheugen inneemt. Dit zorgt daarnaast ook voor een kleinere attack-surface, wat de veiligheid ook ten goede komt.

2.1 *De structuur van Linux-commando's:*

De shell interpreteert de commando's. Alle Linux-commando's hebben een aantal eigenschappen gemeen. Een commando kan zonder opties of parameters gebruikt worden, maar bij veel commando's gebeurt er dan niet veel, of juist te veel. (vb. ***find***)

- **Opties** zijn een extra stukjes informatie die je aan een commando kan meegeven om te zeggen **hoe** je wil dat het commando wordt uitgevoerd. Deze opties zijn vast in het programma geprogrammeerd (een commando is een programma) en je kan dus zelf niet zomaar opties toevoegen. Ze beginnen bijna steeds met een "-".
- **Argumenten** zijn variabele waarden die aan een commando meegegeven worden om het gedrag verder te specifiëren. Meestal gaat het over **waarop** het commando dient toegepast te worden.

Voorbeeld:

ls -h -a -l /home/student

"*ls*" is het commando, "*- h*", "*-a*" en "*-l*" zijn 3 verschillende opties, "*/home/student*" is een argument

Opdracht:

Probeer meteen een aantal Linux-commando-s uit:

who toont de ingelogde gebruikers
whoami toont je eigen loginnaam
date toont de ingestelde datum en tijd

Tip: gebruik de **pijltjestoetsen** ↑ en ↓ om de vorige commando's te herhalen, **TAB** voor tabcompletion. Tab-completion zal je commando, directory, bestand,... aanvullen wanneer dat mogelijk is. Typ bijvoorbeeld *vi* in en druk daarna op de tab-toets. Je krijgt nu alle commando's te zien die met de letters *vi* beginnen. Wanneer er maar 1 mogelijkheid is, vult bash al volledig aan.

3. Hulp in linux

Linux heeft veel bronnen met informatie ingebakken in het operating system, zo ook in de CLI. Hieronder vind je een overzicht van de belangrijkste. Probeer informatie te vinden door al deze commando's eens op elkaar te testen, en op andere commando's en concepten als zoals: mkdir, cd, password, nano,…

3.1 *man*

Van elk Linux-commando kan je de zogenaamde manual (afgekort man, in het Nederlands hanleiding) pages oproepen. Dit doe je door te typen: man *commando*

Zo zal bijvoorbeeld *man ls* de manual page van ls tonen. Wanneer je in de man page zit, kan je naar onderen navigeren door middel van de pijltjestoetsen, page up en page down,... Je kan de man page verlaten door het typen van de letter *q.*

Probeer de man pages zo veel mogelijk te gebruiken wanneer je de werking van een commando niet goed begrijpt. Het zijn de help-files van het Linux-systeem en je kan er zowat alle informatie in terugvinden die nodig zijn voor het gebruik van Linux!

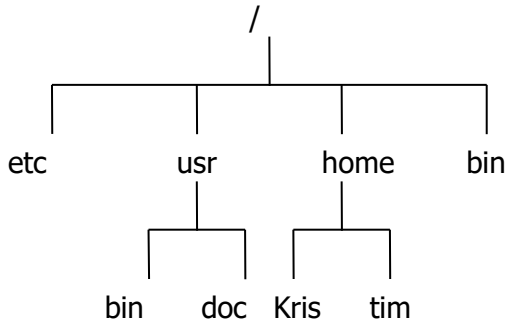
Andere nuttige commando's om informatie te gebruiken zijn info, whatis, en de –help optie.

3.2 *–help*

Veel commando's bevatten een –help optie waarmee je makkelijk en snel een overzicht krijgt van de meest gebruikte parameters en opties van dat commando.

4. Het Linux bestandssysteem

Het Linux-bestandssysteem heeft een vaste structuur, bijvoorbeeld:



Net zoals Windows heeft ook Linux een **bestandssysteem**, een ordening van bestanden in directories en subdirectories die we kunnen voorstellen in een boomstructuur. Er zijn echter wel een aantal belangrijke verschillen tussen Linux en Windows. De root van het Linux bestandssysteem is **/** en niet C:\ of een andere letter. Waar in Windows alle letters een aparte partitie of netwerkfolder weergeven, worden die in Linux allemaal onder dezelfde / geplaatst.

In / zitten een aantal subdirectories, meestal door het systeem vastgelegd. Een subdirectory van / is bijvoorbeeld de directory usr. Van usr is bin dan weer op zijn beurt een subdirectory. We kunnen wisselen van directory met het commando **cd** (change directory). Vanuit / het commando *cd usr/local* ingeven heeft als resultaat dat de huidige werkdirectory nu /usr/local is, hetgeen we kunnen checken met het commando **pwd**.

Opdracht:

Probeer volgende commando's:
cd /usr/local: zorgt ervoor dat we ons naar een andere directory kunnen begeven
cd .. : zal ons 1 directory terug laten keren
cd ../..: zal ons 2 directories terug laten keren **ls** toont de inhoud van de huidige werkdirectory

cd: zal ons terug naar onze homedirectory brengen.
pwd toont de huidige werkdirectory

4.1 *Absoluut of relatief pad*

Het aanduiden van een plaats in deze structuur kunnen we op 2 manieren:

- Met een **absoluut pad:** een absoluut pad is een volledige verwijzing naar een bepaalde directory in het systeem. Een absoluut pad begint dus steeds **vanaf de root /**. Het absoluut pad van de daarnet genoemde directory local is dus /usr/local.
- Met een **relatief pad: d**it is een relatieve verwijzing die begint **vanaf de huidige positie**. Een relatieve verwijzing naar een directory kan korter zijn dan de absolute verwijzing. Wanneer we ons bijvoorbeeld bevinden in de directory /usr kunnen we naar de subdirectory local ook verwijzen gewoon met local. Dit ging ook weer evengoed als met een absoluut pad maar de relatieve verwijzing hier is alleszins korter. Wanneer we een relatief pad gebruiken, kunnen we nog gebruik maken van *2 speciale constructies*:
- Als we in zo een relatief pad ergens **..** gebruiken duidt dat de ouderdirectory aan, de directory die 1 niveau dichter bij ? ligt.
- Gebruiken we ergens **.** duidt dit op de huidige directory.
- Voorbeeld: wanneer we ons bevinden in /usr/local en we willen een relatief pad naar de directory /usr/src is dat relatief pad ../src.

Algemene opmerking*: het is zeer belangrijk deze 2 begrippen goed te begrijpen.* Overal binnen de informatica zul je ze tegenkomen. Bij het gebruik van bestanden of externe bibliotheken bij het programmeren, bij de ontwikkeling van webpagina's, bij hyperlinks, bij het linken van bestanden,… Gebruik maken van Linux zonder te kunnen werken met absoluut en relatief pad is onmogelijk.

Opdracht:

Begeef je naar de directory /usr/local/bin.
Van hieruit willen we naar de directory /usr/src. Dit doen we uiteraard met het commando **cd**. Welke mogelijke argumenten kunnen we aan cd meegeven om in de juiste directory te belanden?

4.2 *Homedirectory*

In het Linux bestandssysteem heeft elke gebruiker (Linux is een multi-user systeem) zijn eigen persoonlijke werkdirectory waar hij en alleen hij toegang tot heeft. Dit noemen we de **homedirectory** van de gebruiker. Ben je ingelogd als guest, dan is je homedirectory /home/guest. Ben je bv de gebruiker joske, dan zal je homedirectory /home/joske zijn.

Als je het commando **cd** gebruikt zonder bijkomend argument, beland je altijd in je homedirectory. Een afkorting die je kan gebruiken in een relatief of absoluut pad naar je homedirectory is **~** (uitgesproken: tilde).

/home/guest/doc is dus hetzelfde als ~/doc (als je als guest bent ingelogd). Je kan door middel van het commando **whoami** opvragen als welke gebruiker je bent ingelogd.

Wanneer je dus vanuit /usr/src naar jouw homedirectory wil, kan je dat weer doen op verschillende manieren.

Opdracht:

Verzin 3 manieren om vanuit /usr/src naar je homedirectory te gaan.

1.....
2.....
3.....

TIPS:
- **Met de pijltjestoetsen kan je je vorige commando's terug oproepen**
- **Bash kent tab-completion, typ eens cd /ho en druk daarna op <tab>, het systeem zal zelf het commando omzetten naar: cd /home**

De inhoud van een directory opvragen doe je met het commando **ls**. Aan dit commando kan je heel wat opties meegeven. Kijk maar eens in de man-page. Let op het verschil tussen opties en argumenten.

Voorbeeld:

```
ls -l /usr/local/bin
```

De -l optie wordt gebruikt om meer details af te beelden.

Als je ls zonder opties of argumenten uitvoert, beeld je gewoon de inhoud af van de directory waarin je je op dat moment bevindt. Doe dit maar eens voor je homedirectory. (onthoud: je kan je homedirectory gaan met het commando *cd* zonder opties of argumenten).

Opdracht

Je ziet enkel de gewone bestanden met het commando ls. Er is een manier om met het commando **ls** ook verborgen bestanden te zien. Vind je ze? Maak gebruik van de help-commando's om de informatie te vinden.
Hoeveel verborgen bestanden zitten er in je homedirectory?
Test ook: *ls /etc/??[ae]** en *ls /etc/n**

Aanmaken van directories kan je doen met het commando **mkdir** met als argument de naam van de nieuwe directory die je wilt aanmaken. Bestanden verwijderen kan je met het commando **rm**. Lege directories verwijderen gebeurt met het commando **rmdir**.

5. De teksteditor vi

Zoals windows notepad heeft om bestanden te editeren, kun je in Linux verschillende teksteditors gebruiken. De editor **vi** is een zeer geavanceerde teksteditor, maar in het begin is het even wennen.

Als je **vi** opstart kom je terecht in **commando mode**. Je kan dus niet direct beginnen te tikken. Tik je het commando **i** in kom je in **insert mode** waarin je tekst kan intikken of **a** voor het de **append mode**. Het verschil zit hem in de positie van de cursor t.o.v. het character dat je typt. Heb je gedaan met tikken kom je terug in commando mode door ESC in te drukken.

Volgende commando's moet je vanbuiten kennen:

Commando	**Actie die uitgevoerd wordt**
:w	sla het bestand op
:w file.txt	sla het bestand op met bestandsnaam file.txt
:wq	sla het bestand op en verlaat vi
:q!	verlaat vi zonder de wijzigingen op te slaan

Good-to-know:

Commando	**Actie die uitgevoerd wordt**
dd	verwijder de lijn waar de cursor zich bevindt
yy	kopieer de huidige regel
5yy	kopieer 5 lijnen vanaf de huidige regel
p	plak (paste) de gekopieerde regels
/zoektekst	Dit zal het woord zoektekst pzoeken. Door te drukken op de letter n zal je naar de volgende plaats waar het woord voorkomt springen.

Opdracht

Typ eerst wat tekst in, zodat je ook daarna kan bewegen in je tekstbestand. Neem bijvoorbeeld volgende tekst over in een bestand tekst.txt:

Mijn naam is …
In deze Linux les, gebruik ik VI als teksteditor

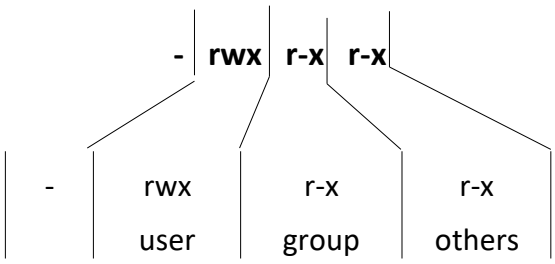
Door de laatste escape kom je nu terug in commando-modus. Probeer nu met de commando's tekst te copy/pasten, verwijderen, enz, ...
Naast vi is er nog een andere zeer populaire texteditor: nano. Deze is iets gebruiksvriendelijker, maar in tegenstelling tot vi niet in alle distributies voorzien. Handig is

de balk onderaan in nano met de belangrijkste opties. Zo kan je daar bijvoorbeeld zien dat om het bestand op te slaan, CTRL-O (van write-out) kan gebruiken. Probeer nano eens met bovenstaande oefening en bemerk de verschillen in werkwijze.

6. Toegangsrechten

Een bestand in Linux heeft 3 soorten toegangsrechten: die van de eigenaar (user of u), die van de groepsgenoten van de eigenaar (group of g) en die van de andere gebruikers (others of o). Ieder van die 3 kan 3 rechten op een bestand/map hebben: leesrechten (read of r), schrijfrechten (write of w) en uitvoerrechten (execute of x). De toegangsrechten van een bestand zie je als je het commando **ls –l** gebruikt. Kijk naar de toegangsrechten van jouw net in vi aangemaakte bestand.

Voorbeeld



In dit voorbeeld heeft de user (de eigenaar) alle rechten: hij mag het bestand zowel lezen, als schrijven (veranderen dus) en uitvoeren. De groepsgenoten en de andere gebruikers mogen enkel lezen en uitvoeren; niet wijzigen (vandaar het streepje waar anders de w zou staan).

De toegangsrechten veranderen kan je met het bevel **chmod**. Als argumenten bij chmod geef je het recht dat je wil wijzigen en de bestandsnaam.

		r	w	x
4 staat voor r	4	1	0	0
2 staat voor w	2	0	1	0
1 staat voor x	1	0	0	1

Zoals je kan opmerken komen deze getallen overeen met de binaire waarde van de positie van dat getal. Je kan deze dus ook gewoon optellen per groep (u, g of o) als zijnde binaire getallen. Om -rwxr-xr-x te maken van een bestand is het commando dus **chmod 755 bestand**.

Een alternatieve manier om permissies te plaatsen is werken met de letters horende bij een rechtengroep, en dan het + of – character om het recht aan te passen. Bijvoorbeeld: **chmod u-w,g+w,o+w bestand** neemt het schrijfrecht af van de eigenaar, geeft de groepsgenoten schrijfrechten en doet hetzelfde met de andere gebruikers voor het bestand **bestand**.

Opdracht

De toegangsrechten van je bestand tekst.txt moeten worden veranderd naar -rwxr-xr--.
Dat kan op 2 manieren. Welke? En doe dat ook op 1 van die manieren en controleer.

7. Nog enkele commando's

We bespreken hier nog enkele bijkomende commando's, vooral voor het bestandssysteem.
Met het commando **cp** kunnen we bestanden kopiëren. Het commando heeft 2 argumenten:

cp bronbestand doelbestand

Het commando **mv** gebruiken we om bestanden te verplaatsen. Verplaatsen heeft dus als gevolg dat het bronbestand verdwijnt.

mv bronbestand doelbestand

Voor beide commando's geldt dat zowel het doelbestand als het bronbestand ook een directory mag zijn.

Het commando **rm** laat toe om bestanden te verwijderen.

Voorbeeld

cp file1 file2 kopieert het bestand file1 naar het bestand file2. file1 blijft dus bestaan

cp file1 /home/guest/ kopieert het bestand file1 naar de directory /home/guest.

Opdracht

Maak een bestand opdracht.txt en een subdirectory aan in jouw persoonlijke directory met als naam opdracht. Verplaats het bestand opdracht.txt naar deze directory. Kopieer daarna dat bestand weer terug naar de oorspronkelijke directory. Verwijder nu de directory opdracht. Verwijder ook het originele bestand opdracht.txt.

Noot: Indien je een directory met nog andere bestanden of directories erin wil verwijderen, zal je een extra optie moeten gebruiken. Gebruik de help-mogelijkheden van bash om deze te vinden.

Verborgen bestanden

Bestanden waarvan de naam met een '.' begint zijn verborgen bestanden. Met het commando ls zal je ze niet zien, tenzij je de optie –a gebruikt.

Opdracht

Probeer eens een verborgen bestand te maken, dit kan bijvoorbeeld op volgende manieren:

1. vi .verborgen
2. vi

3. en verder het bestand opslaan als “.onzichtbaar”
4. een bestaand bestand een andere naam geven “.verdwenen”

(dit doe je met het commando mv – verplaatsen met een andere naam)