



BACHELOR TOEGEPASTE INFORMATICA

Linux Command Line

Computer Systems

2.0

Versie

Roel Standaert

Auteur(s)

HISTORIEK

Datum	Versie	Omschrijving	Auteur
27/09/2025	2.0	Update voor academiejaar 2025-2026.	Roel Standaert
29/09/2024	1.0	Eerste versie	Roel Standaert

INHOUD

5 LINUX COMMAND LINE	4
5.1 INTRODUCTIE	4
5.2 TIPS VOOR EFFICIËNT SHELL-GEBRUIK	6
5.3 SHELL-COMMANDO'S	8
5.4 DOCUMENTATIE	10
5.5 NAVIGEREN	13
5.6 MAPPEN EN BESTANDEN BEHEREN	18
5.7 ENKELE BASISCOMMANDO'S SAMENGEVAT	21
5.8 VARIABELEN	25
5.9 TEKST EDITOR	26
5.10 SOFTWAREBEHEER	27
5.11 RECHTEN	32
5.12 ZOEKEN	39
5.13 PIPES	41
5.14 OUTPUT REDIRECTION	44
5.15 TEKST TRANSFORMEREN	45
5.16 GEcombineerde oefening	47

5 Linux command line



1. De student kent de concepten van automatisatie, en kan deze implementeren voor de automatisatie van eenvoudige beheerstaken.
2. De student kan vlot gebruik maken van verschillende besturingssystemen, zowel in een desktopomgeving als in een command line interface.

5.1 Introductie

Linux distributies (en andere UNIX-aanverwanten) zijn alomtegenwoordig in de IT-sector, voornamelijk op servers. Op deze servers wordt vaak de backend gerund van een applicatie. Het beheer van deze servers gebeurt typisch zonder een grafische interface, maar eerder via de command line. Het loont dus om een basis command line gebruik onder de knie te hebben.

In dit hoofdstuk zal je verschillende oefeningen tegenkomen. Je kan er dus best een Linux terminal bij nemen. Leren werken met de command line doe je vooral door het zelf te doen, niet door erover te lezen. Gebruik bijvoorbeeld de virtuele machine die je hebt geïnstalleerd in het virtualisatie-labo.

5.1.1 Waarom de command line gebruiken?

Je kan je afvragen waarom je de command line zou gebruiken, als grafische interfaces zoveel gebruiksvriendelijker lijken. In deze sectie zullen we enkele voordelen van de command line bekijken.

Snelheid

Een ervaren command line gebruiker kan bepaalde handelingen vaak sneller uitvoeren dan die met een GUI mogelijk zijn. Als je het commando kent om iets te

doen heb je vaak veel sneller de terminal geopend (CTRL+ALT+T op Ubuntu) en het commando getypt, dan als je door menu's moet zoeken naar de juiste optie.



Probeer op een Ubuntu virtuele machine eens te achterhalen wat je IP-adres is via de grafische interface. Probeer daarna eens hetzelfde te doen in de terminal: open de terminal (CTRL+ALT+T en gebruik het commando `ip addr`. Wat ging sneller?

Zeker als de handelingen complexer worden, zoals “verplaats alle bestanden die eindigen in `.mp4` van de Downloads-map naar de Videos-map in jouw home folder, kan dit heel snel met het volgende commando:

```
mv Downloads/*.mp4 Videos/
```

Je kan shell-commando's een beetje vergelijken met keyboard shortcuts: je kan sneller iets kopiëren en plakken met CTRL+C en CTRL+V dan dit te doen met de muis.

Efficiëntie

Een GUI (*Graphical User Interface*) belast een systeem veel harder (gebruikt meer CPU en meer geheugen) dan een CLI (*Command Line Interface*). Als je het meeste uit je systeem wil halen en een GUI niet strikt noodzakelijk is, dan kan je die even goed weglaten. Vandaar hebben Linux servers typisch geen grafische interface.

Automatisatie

Als je bepaalde commando's vaak samen moet uitvoeren is dit heel snel geautomatiseerd: je plaatst deze commando's gewoon in een script.

Taken automatiseren bij een GUI is typisch moeilijker: daarbij moet er gebruik moeten worden gemaakt van macro's die bepaalde toetsencombinaties uitvoeren of die automatisch klikken op opties. Dit kan dan ook niet altijd in de achtergrond, en deze automatisatiemethodes zijn vaak gevoelig aan kleine veranderingen in de GUI.

Meer controle

Bij een GUI ben je beperkt tot de opties die de grafische interface aanbiedt. Als een optie niet gemakkelijk voorzien is in de interface, dan ga je misschien moeten zoeken in complexe menu's, wizards of instellingen. Veel shell-commando's hebben tal van opties die het commando zo aanpassen zodat het exact doet wat je wil. Doet één commando niet wat je wil doen? Dan kan je meerdere commando's gaan combineren. Hoe dat moet zie je nog later in sectie 5.11.4 .

Eenvoudige implementatie

Er steekt veel ontwikkelingstijd in het maken van een GUI. Een command line interface is typisch sneller geïmplementeerd. Vandaar dat verschillende tools die je als informaticus zal tegenkomen enkel een command line interface hebben.

Remote access

Een desktopomgeving kan je accessen via een remote desktopapplicatie, maar dit gaat typisch trager en wordt niet handig als je meerdere system tegelijk wil beheren. Via de command line en SSH (*Secure Shell*) kan je gemakkelijk inloggen op meerdere servers tegelijk.

5.2 Tips voor efficiënt shell-gebruik

Met de shell leren werken kan frustrerend zijn als je telkens opnieuw dezelfde commando's moet typen, er telkens een klein foutje in je commando zit, enz. Deze tips helpen je om iets efficiënter met de shell te werken.

5.2.1 TAB completion

Het lijkt zeer vermoeiend om telkens opnieuw volledige commando's of bestandsnamen te typen, bovendien is een typfout snel gemaakt. Ervaren shell-gebruikers typen daarom zelden volledige commando's zelf. Typ gewoon de eerste paar karakters en gebruik de TAB-toets om het commando te vervolledigen.

Zijn er meerdere opties? Dan zal één keer op de TAB-toets drukken niets doen. Als je echter tweemaal op TAB drukt zal je de verschillende opties te zien krijgen.



Probeer eens zelf:

1. Open de terminal (CTRL+ALT+T).
2. Typ `cd Do` en druk tweemaal op TAB. Je zal zien dat er twee opties zijn: Documents en Downloads.
3. Maak er `cd Dow` van en druk nog eens op TAB. Je zal zien dat dit vervolledigd wordt naar `cd Downloads/`

5.2.2 History en command recall

De shell (in ons geval *bash*) houdt je vorige commando's bij. Wil je het vorige commando opnieuw uitvoeren, of was het niet helemaal correct en wil je het aanpassen? Gebruik de pijltjestoetsen (omhoog en omlaag) om door de vorige commando's te gaan. Als je het juiste commando gevonden hebt kan je dat aanpassen en uitvoeren.

Voer je vaak hetzelfde commando uit? Gebruik CTRL+R om snel door jouw vorige commando's te zoeken. Typ een deel van het commando en je zal het zien verschijnen.



Probeer eens zelf:

1. Open de terminal (CTRL+ALT+T).
2. Voer het commando `ls -la /etc` uit.
3. Druk op de pijltjestoets naar boven om dit commando terug te zien.
4. Pas het commando aan naar `ls -la /usr/bin`, en voer dat commando uit.
5. Gebruik CTRL+R en typ `e`. Je zal het commando `ls -la /etc` terug zien verschijnen. Druk op ENTER om het uit te voeren.

5.2.3 CTRL+C en CTRL+D

Blijft een commando “hangen”? Dit kan zijn omdat het input verwacht¹, omdat het lang duurt om het commando uit te voeren² of omdat het een commando is dat bedoeld is om te blijven draaien³. Je moet dan je terminal niet sluiten. Typisch kan het programma dan gewoon stoppen met **CTRL+C**.

Andere commando's zoals `python3` of de shell zelf (`bash`) starten een interactieve shell. Meestal kan je **CTRL+D** gebruiken om deze shell te verlaten.



Probeer eens zelf: open de terminal (`CTRL+ALT+T`) en voer het `cat`-commando uit. Gebruik `CTRL+C` om dit commando te stoppen. Sluit vervolgens de terminal met `CTRL+D`.

5.3 Shell-commando's

Wat is nu eigenlijk een shell-commando? Er zijn eigenlijk twee verschillende soorten:

- Uitvoerbare bestanden
- Built-in commando's

5.3.1 Uitvoerbare bestanden

De meeste shell-commando's zijn gewoon uitvoerbare bestanden (of programma's). Als je een commando zoals `ls` wil uitvoeren, dan zal `bash` dat opzoeken in de `PATH`-variabele. Bekijk de `PATH`-variabele eens:

```
echo $PATH
```

¹ Dit kan bijvoorbeeld als je het `cat`-commando gebruikt zonder inputbestanden.

² Bijvoorbeeld een groot bestand kopiëren.

³ `python3 -m http.server` start bijvoorbeeld een HTTP-server in de huidige map.

Je zal iets zien dat lijkt op:

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
```

Dit zijn de paden (gescheiden door een “.”) die doorzocht zullen worden als je een commando uitvoert. Om `ls` uit te voeren probeert de shell deze paden één voor één:

1. Is `/usr/local/sbin/ls` een uitvoerbaar bestand? → Nee
2. Is `/usr/local/bin/ls` een uitvoerbaar bestand? → Nee
3. Is `/usr/sbin/ls` een uitvoerbaar bestand? → Nee
4. Is `/usr/bin/ls` een uitvoerbaar bestand? → Ja, voer `/usr/bin/ls` uit.

Je kan voor elk commando checken waar het uitvoerbaar bestand zich bevindt met `which`. `which ls` geeft bijvoorbeeld `/usr/bin/ls`.

5.3.2 Built-in commando's

Je zal merken dat als je `which cd` probeert uit te voeren dat je geen output krijgt. Dat komt omdat `cd` een built-in commando is: het verandert de map waarin de shell zich momenteel bevindt. Dit kan niet geïmplementeerd worden met een uitvoerbaar bestand. Sommige commando's bestaan zowel als uitvoerbaar bestand (`/usr/bin/echo`) als als een built-in commando. Dit wordt vaak gedaan omdat het efficiënter is om dit in de shell in te bouwen dan om dat als een apart proces op te starten. De shell zal hierbij de voorkeur geven aan het built-in commando.

5.3.3 De structuur van een commando

Je kan aan een commando verschillende opties en argumenten meegeven. Je kan zo dus eigenlijk een commando (zoals `cat -n /etc/passwd`) opdelen in drie stukken:

1. Het commando zelf (`cat`). Dit kan een uitvoerbaar bestand zijn of een built-in commando.
2. De opties (`-n`). Dit past het gedrag van een commando aan. De `-n`-optie van `cat` zorgt ervoor dat er lijnummers worden uitgeprint.

3. De argumenten (/etc/passwd). Deze geven extra informatie door aan het commando. Bij het commando cat zijn dit de bestanden die getoond moeten worden.



Merk op dat spaties van belang zijn in de shell: commando's en hun argumenten worden met behulp van spaties gescheiden. Vergeet ze dan ook niet!

Bijvoorbeeld: `cat -n` voert cat uit met de optie -n, `cat -n` geeft "command not found".

Staan er spaties in een argument? Gebruik dan aanhalingstekens (enkele (" ' ") of dubbele (" " ")) of *escape* de spaties met een backslash: `cat 'bestand met spaties'`, `cat "bestand met spaties"` of `cat bestand\ met\ spaties`.

5.4 Documentatie

Als je aan een Linux-veteraan vraagt hoe je een bepaald commando moet uitvoeren is de kans groot dat je als antwoord krijgt: RTM: Read The Manual.⁴ Elke Linux-distributie (en elke UNIX-variant) komt met een ingebouwde handleiding. Het loont om deze te leren gebruiken.

5.4.1 man

De *manpages* bevatten documentatie over allerlei aspecten van jouw besturingssysteem: niet alleen commando's staan erin maar ook documentatie over bijvoorbeeld configuratiebestanden.

⁴ Als die wat slechtgezind is zal die daar nog een F aan toevoegen.

Je kan de manual raadplegen met het commando `man`. Als je bijvoorbeeld wil uitzoeken hoe je `ls` kan gebruiken:

```
man ls
```

Je kan hier op verschillende manieren door scrollen:

1. Met de muis
2. Met de pijltjestoetsen
3. Met Page Up en Page Down
4. Met `d` (down, halve pagina naar beneden) en `u` (up, halve pagina naar boven)

Je kan ook zoeken door een slash ("`/`") te typen gevolgd door jouw zoekterm.

Bevestig met Enter. Je kan dan door de zoekresultaten gaan met `n` (next, volgende match) en `N` (SHIFT+N, vorige match). De manual sluiten doe je met `q`.



Zoek eens uit met `man` hoe je in de output van `ls -l` ervoor kan zorgen dat de grootte van bestanden (de vierde kolom) in een "human-readable" formaat wordt getoond.

Secties

De manual pages zijn ingedeeld in secties. Alle programma's bevinden zich in sectienummer 1. Als je expliciet een bepaalde sectie wil raadplegen zal je dit als argument moeten meegeven: `man 1 ls` zoekt `ls` op in sectie 1.⁵

⁵ Wil je meer informatie over hoe je `man` moet gebruiken en welke secties er allemaal zijn? Gebruik het commando `man man`!

5.4.2 help

Built-in commando's hebben geen manpage. Deze zal je moeten raadplegen met help:

```
help cd
```



Zoek eens uit met `help` welke optie je aan `echo` moet geven om een string te outputten zonder newline op het einde.

5.4.3 apropos

Weet je niet exact welk commando je nodig hebt? Je kan zoeken in alle manpages met het commando `apropos`. Als je expliciet wil zoeken in een bepaalde sectie kan je het sectienummer meegeven, bijvoorbeeld om de zoekterm "zoekterm" te zoeken in sectie 1:

```
apropos -s 1 zoekterm
```



Zoek eens uit met `apropos` welke commando's je kan gebruiken om een tijdelijk bestand (*temporary file*) aan te maken.

5.4.4 --help

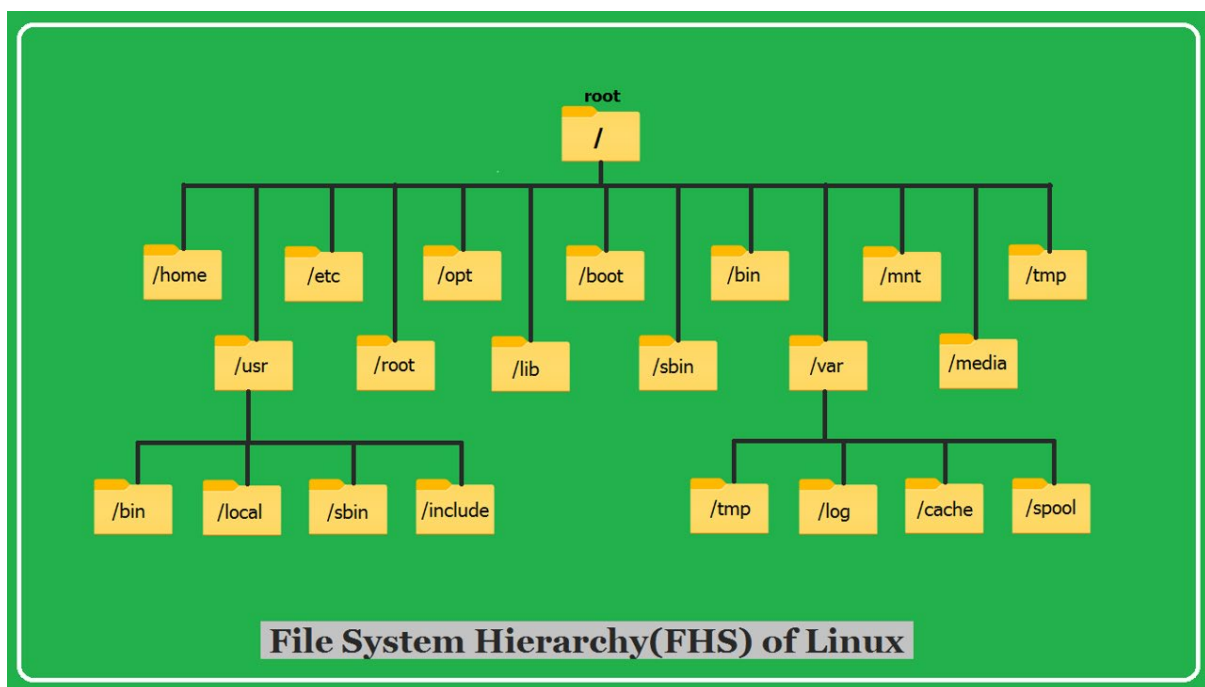
Tenslotte zit een overzicht van een commando en de verschillende opties vaak ingebouwd in het commando zelf. Aan veel commando's kan je `--help` toevoegen om dit overzicht te krijgen. Soms kan je dit ook afkorten tot `-h` of `-?`. Deze afkorting kan spijtig genoeg sterk verschillen van commando tot commando.

5.5 Navigeren

Eén van de meest basic vaardigheden die je onder de knie moet hebben als je de Linux shell wil gebruiken is hoe je moet navigeren: hoe ga ik naar de juiste map op mijn bestandssysteem en waar kan ik deze vinden?

5.5.1 File System Hierarchy (FSH)

Linux (en andere UNIX-derivaten) plaatst alle bestanden en mappen op jouw systeem in een boomstructuur beginnende bij de *root* directory (figuur 4.1), aangegeven met een slash ("/"). Bij Linux gebruiken we dus geen schijfletters zoals op Windows: er is geen C-drive, D-drive, ... Als er meerdere harde schijven of partities zijn worden die ergens geplaatst ("gemount") in die boomstructuur (meer daarover in een later hoofdstuk).



Figuur 5-1: De Linux File System Hierarchy

Verschillende mappen in de Linux File System Hierarchy hebben een vaste functie:

- /etc bevat configuratiebestanden
- /home bevat de *home directories* van de gebruikers: /home/bassie bevat de bestanden van de gebruiker bassie en /home/adriaan bevat de bestanden van de gebruiker adriaan.

- /var/log bevat logbestanden
- /usr/bin bevat uitvoerbare bestanden (programma's)
- ...

5.5.2 Working directory

In de shell (of eender welk programma) wordt er bijgehouden in welke map je je momenteel bevindt. Dit noemt men de *working directory*. Je kan zien wat je huidige pad is met `pwd` (*print working directory*).



Open een nieuwe terminal (CTRL+ALT+T) en gebruik het `pwd` commando. Het resultaat is jouw *home directory*. Als je een nieuwe shell opent, vertrek je altijd hier.

5.5.3 Wisselen tussen directories

Je gebruikt de shell built-in `cd` (*change directory*) om van directory te wisselen. Als je bijvoorbeeld naar de `/etc`-map wil gaan kan je `cd /etc` gebruiken. Je kan altijd terugkeren naar jouw home directory door het `cd`-commando zonder argumenten te gebruiken.

5.5.4 Paden

Er zijn twee soorten paden:

- Relatieve paden
- Absolute paden

Relatieve paden

Een **relatief pad** vertrekt vanuit de huidige working directory. Je kan de bovenliggende map aangeven met twee puntjes (`..`). Als je je in je home directory bevindt (`/home/username`), dan gaat `cd ..` naar de `/home` map.

De huidige map kan je aangeven met één puntje ("."). Als je in je home directory bevindt, dan zal `ls .` je de inhoud tonen van de huidige map (dit geeft natuurlijk hetzelfde als `ls` zonder argumenten).

Een aantal voorbeelden van het gebruik van relatieve paden:

- `cd Documents` → verandert de working directory naar de Documents-map
- `cd ../` → verandert de working directory naar de bovenliggende map
- `cd ../../` → gaat twee directories omhoog
- `cd ../../etc` → gaat twee directories omhoog, en dan naar de etc-map

Absolute paden

Een **absoluut pad** vertrekt van de *root directory* ("/"). Een absoluut pad begint dus altijd met een slash.

Tilde

Er is één karakter met een speciale betekenis: de tilde (~). Je kan dit gebruiken om naar je home directory te verwijzen. De volgende 3 commando's doen dus hetzelfde:

```
cd
cd ~
cd /home/username
```

Als je je eender waar bevindt kan je dus bijvoorbeeld gemakkelijk naar de Documents map in je home directory gaan met:

```
cd ~/Documents
```

De shell zal de ~ vervangen door je home directory om zo een absoluut pad te bekomen.

Oefeningen

Probeer de volgende oefeningen eens uit in je terminal.



Oefening 1:

1. Open de “Text Editor” in je grafische omgeving. Schrijf wat tekst naar keuze en sla dit op als een tekstbestand op je bureaublad (Desktop).
2. Probeer nu de inhoud van het bestand te tonen in je terminal met het cat-commando.



Oefening 2: Toon de inhoud van het bestand `/etc/lsb-release` door gebruik te maken van één enkel commando.



Oefening 3: Schrijf een oneliner die naar de map `/tmp` navigeert, daar een leeg bestand aanmaakt met de naam `HelloWorld.txt` (dit kan met het commando `touch`), en vervolgens terug naar de home directory van de user navigeert.



Een oneliner is een combinatie van verschillende commando's in één lijn. De gemakkelijkste vorm om verschillende commando's aan elkaar te koppelen is door een `&&` ertussen te plaatsen. `&&` wil zeggen: voer het volgende commando uit als het vorige commando geslaagd is. Bijvoorbeeld:

```
mkdir NieuweMap && cd NieuweMap
```

Dit maakt een map genaamd `NieuweMap` aan (`mkdir` = make directory) en gaat daarna naar deze nieuwe map.

De oplossingen vind je op de volgende pagina.

Oplossingen

Oefening 1

Als je jouw bestand bijvoorbeeld `MijnBestand.txt` hebt genoemd, dan kan je bijvoorbeeld dit commando gebruiken:

```
cat ~/Desktop/MijnBestand.txt
```

Oefening 2

Net als in de vorige oefening kunnen we het `cat`-commando gebruiken. `/etc/lsb-release` is een absoluut pad (begint met een `/`) dus we kunnen dit vanuit eender welke map uitvoeren:

```
cat /etc/lsb-release
```

Oefening 3

Je kan `&&` gebruiken om 2 `cd`-commando's en één `touch`-commando aan elkaar te hangen:

```
cd /tmp && touch HelloWorld.txt && cd
```

5.6 Mappen en bestanden beheren

5.6.1 Bestanden maken en verwijderen

Je bent hiervoor al de commando's `cat` en `touch` tegengekomen. Met `cat` kan je de inhoud van een bestand zien en met `touch` kan je een nieuw leeg bestand aanmaken.⁶ Als je een bestand wil verwijderen gebruik je `rm` (**r**emove).

Om dus een bestand genaamd `MijnBestand.txt` wil aanmaken in de huidige map gebruik je:

```
touch MijnBestand.txt
```

Om dit dan terug te verwijderen gebruik je:

```
rm MijnBestand.txt
```

5.6.2 Directories maken en verwijderen

Met het commando `ls` kunnen we de inhoud van een map zien. Als we een nieuwe map willen aanmaken kan dat met `mkdir`. Om dus een map genaamd `newdir` aan te maken in de huidige directory gebruik je dus:

```
mkdir newdir
```

`mkdir` komt van "**m**ake **d**irectory".

Om een map te verwijderen gebruik je `rmdir` (**r**emove **d**irectory):

```
rmdir newdir
```

`rmdir` zal enkel mappen verwijderen die leeg zijn en weigeren om mappen te verwijderen waar iets in zit. Om een map met de volledige inhoud ineens te

⁶ Dit is eigenlijk een bijwerking van het `touch`-commando. `touch` dient om de datum en tijd waarop een bestand laatst gewijzigd is te updaten naar het huidige tijdstip. Als het bestand echter niet bestaat, wordt het nieuw aangemaakt. Deze bijwerking is uiteindelijk waar `touch` het vaakst voor gebruikt wordt.

verwijderen kan je `rm` met de extra opties `-r` en `-f` gebruiken (je kan die combineren tot één optie `-rf`), bijvoorbeeld:

```
rm -rf newdir
```



Verschillende commando's hebben een `-r` of een `-R` optie die betekent "recursief". **Recursief** wil zeggen dat dezelfde operatie herhaaldelijk wordt toegepast. `rm -r` past dus `rm` toe op zowel de gegeven map, als alle onderliggende mappen, als alle mappen daaronder, enzovoort.



De `-f` optie betekent "force". Dit wil zeggen dat er niet nog eens om een bevestiging wordt gevraagd. `rm` zal dit in bepaalde gevallen doen, vandaar dat je meestal de twee opties samen ziet als één optie: `-rf`.

5.6.3 Kopiëren en verplaatsen

Twee andere onmisbare commando's zijn `cp` (**copy**) en `mv` (**move**). Hiermee kan je een bestand kopiëren en plakken of knippen en plakken naar een andere locatie. Zo kopieert het volgende commando het bestand `/etc/passwd` naar de map `/home/username`:

```
cp /etc/passwd /home/username/
```

Het oorspronkelijke bestand blijft behouden. Als we `mv` zouden gebruiken in de plaats van `cp` zou dat niet het geval zijn.

Je kan hier ook relatieve paden gebruiken. Als we bijvoorbeeld `cp /etc/passwd Documents` uitvoeren, zal het bestand rechtstreeks naar `/home/username/Documents` gekopieerd worden. Als we `cp /etc/passwd .` zouden uitvoeren wordt het naar de huidige working directory gekopieerd.

In alle bovenstaande commando's kan je `cp` door `mv` vervangen. Het resultaat zal zijn dat het oorspronkelijke bestand verwijderd wordt. Met deze commando's

kunnen we ook de naam van een bestand wijzigen. Neem bijvoorbeeld het volgende commando:

```
mv /home/username/passwd Downloads/password_file.txt
```

Dit commando neemt het bestand dat we net gekopieerd hebben en verplaatst het naar de map Downloads. Omdat we dit keer een bestandsnaam meegeven wordt ook de naam van het bestand gewijzigd.

Je kan cp en mv ook gebruiken om gehele mappen te kopiëren of te verplaatsen. Bij cp zal je wel de optie -r (recursief) moeten gebruiken.

5.7 Enkele basiscommando's samengevat

Onderstaande tabel geeft een overzicht van enkele basiscommando's:

Commando	Betekenis
<code>echo <i>string</i></code>	Toon de string <i>string</i>
<code>cd <i>path</i></code>	Verander de working directory naar <i>path</i>
<code>pwd</code>	Toon de huidige working directory
<code>ls</code>	Toon de inhoud van de working directory
<code>ls <i>path</i></code>	Toon de inhoud van de <i>path</i> directory
<code>touch <i>path</i></code>	Maak een leeg bestand aan op <i>path</i>
<code>cat <i>path</i></code>	Toon de inhoud van het bestand op <i>path</i>
<code>mkdir <i>path</i></code>	Maak de map <i>path</i> aan
<code>rm <i>path</i></code>	Verwijder het bestand <i>path</i>
<code>cp <i>from to</i></code>	Kopieer het bestand <i>from</i> naar <i>to</i>
<code>mv <i>from to</i></code>	Verplaats het bestand of de map <i>from</i> naar <i>to</i>

Op Toledo zal je ook verschillende *cheatsheets* vinden met veelgebruikte commando's.

Oefeningen

Probeer de volgende oefeningen eens uit in de terminal. Kijk naar de manpages om te weten welke eventuele opties je nodig hebt:



Oefening 1: Vertrek vanuit je home directory. Ga naar de Documents-map. Maak daar een leeg bestand genaamd "mijn bestand". Let op de spaties! Kijk na dat je dit bestand hebt gemaakt.



Oefening 2: Vertrek vanuit je home directory. Toon in één commando de inhoud van het bestand `passwd` in de `/etc`-map via een **relatief pad**. Verander niet van working directory!



Oefening 3: Vertrek vanuit je home directory. Geef met één commando een overzicht van alle bestanden onder `/etc`, ook de bestanden in de onderliggende mappen (tip: gebruik `man` of `--help`).



Oefening 4: Vertrek vanuit je home directory. Kopieer met één commando het bestand `/etc/group` naar de map `Documents` onder je home directory en hernoem het naar `groups.txt`.



Oefening 5: Vertrek vanuit je home directory. Maak een map genaamd `dir1` en ga naar deze map. Maak daarin in één commando drie bestanden aan: `file1`, `file2` en `file3`. Ga terug naar je home directory. Kopieer in één commando de volledige inhoud van `dir1` naar een nieuwe map genaamd `dir2`.

De oplossingen vind je op de volgende pagina.

Oplossingen

Oefening 1:

1. Als je je nog niet in je home directory bevond, gebruik dan `cd` om ernaartoe te gaan.
2. Ga naar de Documents-map: `cd Documents` (typ `cd Doc` gevolgd door TAB om het jezelf gemakkelijk te maken).
3. Maak het bestand "mijn bestand": `touch "mijn bestand"` (`touch mijn bestand` maakt twee bestanden: `mijn` en `bestand`).
4. Kijk na: `ls`

Oefening 2:

1. Als je je nog niet in je home directory bevond, gebruik dan `cd` om ernaartoe te gaan.
2. Toon de inhoud van `/etc/passwd` met een **relatief pad**: `cat ../../etc/passwd`. De huidige map is `/home/username`, dus ga je twee mappen naar boven om naar `/` te gaan, daarna kan je afdalen naar `etc/passwd`. Met een absoluut pad kan je gewoon `cat /etc/passwd` doen.

Oefening 3:

1. Als je je nog niet in je home directory bevond, gebruik dan `cd` om ernaartoe te gaan.
2. Als je `ls --help` gebruikt of man `ls` zal je een optie `-R` tegenkomen met de beschrijving "list subdirectories recursively". Recursief wil zeggen dat dezelfde operatie herhaaldelijk wordt toegepast. Dit past dus `ls` toe op zowel de huidige map, als alle onderliggende mappen, als alle mappen daaronder, enzovoort.
3. Het resulterende commando is dus: `ls -R /etc`

Oefening 4:

1. Als je je nog niet in je home directory bevond, gebruik dan `cd` om ernaartoe te gaan.
2. Om te kopiëren gebruik je het commando `cp`. Voor `/etc/group` gebruiken we best het absolute pad, voor Documents kunnen we een relatief pad gebruiken (we zitten namelijk al in de home directory). Het resulterende commando is dus: `cp /etc/group Documents/group.txt`.

Oefening 5:

1. Als je je nog niet in je home directory bevond, gebruik dan `cd` om ernaartoe te gaan.
2. Maak een map genaamd `dir1`: `mkdir dir1`.
3. Ga naar deze map: `cd dir1`
4. Maak in één commando drie bestanden aan: `touch file1 file2 file3`.
Door de manpage van `touch` te raadplegen vind je dat je meerdere bestandsnamen kan geven aan `touch`. Dit wordt aangegeven door `FILE...` (let op de drie puntjes):

NAME

`touch - change file timestamps`

SYNOPSIS

`touch [OPTION]... FILE...`

5. Ga terug naar je home directory: `cd` of `cd ..`
6. Kopieer `dir1` naar `dir2`: `cp -r dir1 dir2`. In de manpage van `cp` vind je de `-r`-optie: “copy directories recursively”.

5.8 Variabelen

In de shell, net als in programmeertalen, kan je gebruik maken van variabelen. Eerder heb je bijvoorbeeld al eens de waarde van de PATH-variabele bekeken. Als we een waarde willen toekennen aan een variabele kunnen we dat als volgt doen:

```
variabele=waarde
```

Hier wordt de waarde *waarde* toegekend aan de variabele met naam *variabele*.



Merk op dat er geen spatie staat tussen de naam van de variabele, de = en de waarde. Dit is belangrijk. Als je `variabele =waarde` zou typen, dan wordt `variabele` geïnterpreteerd als een commando en niet als de naam van een variabele. Als je `variabele= waarde` zou typen, dan wordt dit geïnterpreteerd als: zet `variabele` gelijk aan de lege string (`""`) en voer daarna het commando `waarde` uit.

Als er spaties in de waarde staan zal je gebruik moeten maken van aanhalingstekens of escapes:

```
lector='Roel Standaert'
```

Met het dollarteken (\$) kan je de waarde uit een variabele halen. Als je bijvoorbeeld de variabele `lector` hebt met waarde "Roel Standaert", zal dit "Roel Standaert" outputten:

```
echo $lector
```



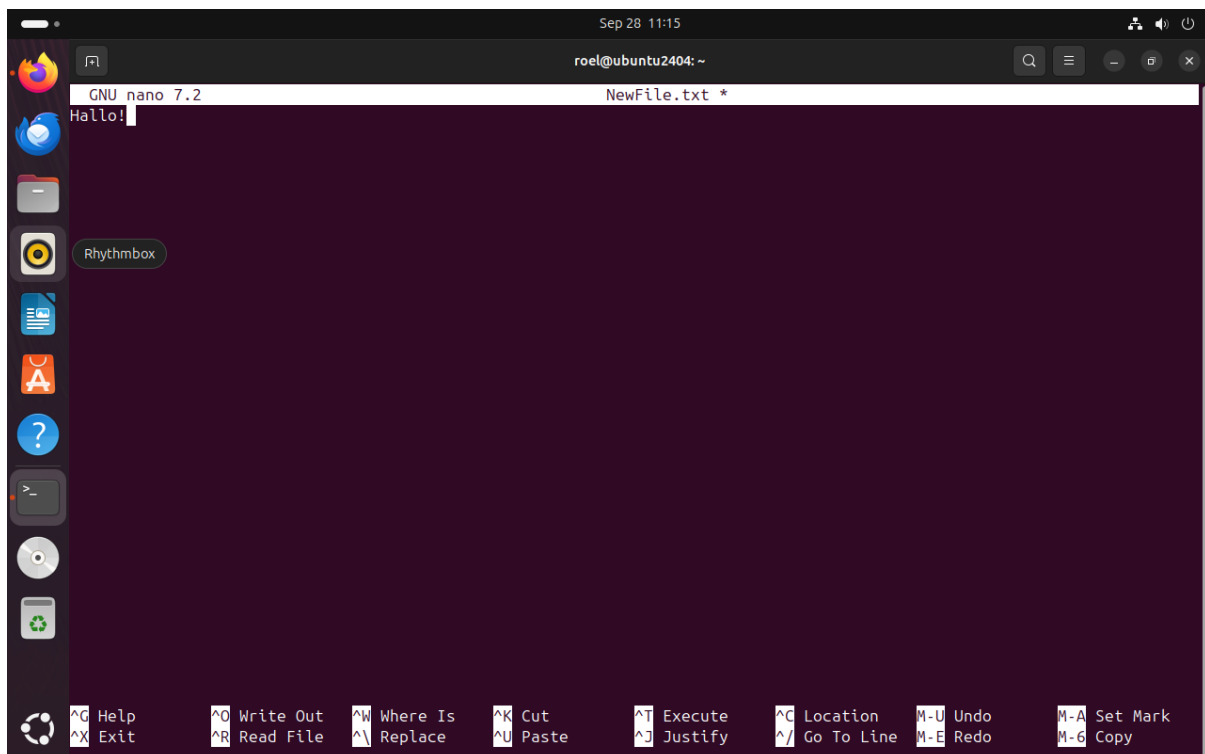
`PATH` is een speciale soort variabele genaamd een omgevingsvariabele (*environment variable*). Je kan alle omgevingsvariabelen bekijken met het commando `env`.

5.9 Tekst editor

Het is natuurlijk heel handig om bestanden te kunnen aanmaken, kopiëren en verwijderen, maar wat als we de inhoud willen aanpassen via de command line? De gemakkelijkste manier om dit te doen is met een CLI-teksteditor zoals nano of vim.

Vim is een heel krachtige bewerker, maar is niet zo geschikt voor beginnende Linux-gebruikers. Vandaar dat we nano gaan leren gebruiken.

Met het commando `nano NewFile.txt` kan je een teksteditor openen en beginnen te schrijven in het bestand `NewFile.txt`. Als dat bestand nog niet bestaat zal het automatisch aangemaakt worden. Hier kan je gewoon beginnen typen, aanpassen en verwijderen zoals je met eender welke teksteditor doet. Je kan uiteraard je muis niet gebruiken om je tekstcursor te verplaatsen: je gebruik moeten maken van de pijltjestoetsen.



Je zal zien dat er onderaan opties staan zoals `^O` voor “Write Out” (opslaan) en `^X` voor “Exit”. `^` wordt hier gebruikt om de CTRL-toets aan te geven. Je kan je bestand dus opslaan met `CTRL+O` en nano verlaten met `CTRL+X`.

5.10 Softwarebeheer

Natuurlijk willen we met ons systeem meer gaan doen dan enkel bestanden aanmaken en verplaatsen. Net zoals je in Windows-systemen software kan installeren met `.exe` of `.msi` bestanden, kan je in Linux-systemen software installeren met verschillende soorten pakketten (bijvoorbeeld `.deb` of `.rpm`), maar de meest courante manier om software te installeren door gebruik te maken van een package manager.

5.10.1 Package manager

Ubuntu maakt, net zoals veel Linux distributies, gebruik van een package manager. In het geval van Ubuntu is dit APT (Advanced Packaging Tool), aangestuurd door het commando `apt`.

Om een eerste idee te geven van hoe het in zijn werk gaat, probeer eens het commando `apt list` uit te voeren. Hier zal je een grote lijst krijgen van alle softwarepakketten die je kan installeren. Je zal merken dat dit een vrij lange lijst is, op het moment van schrijven zijn dit 94662 pakketten.

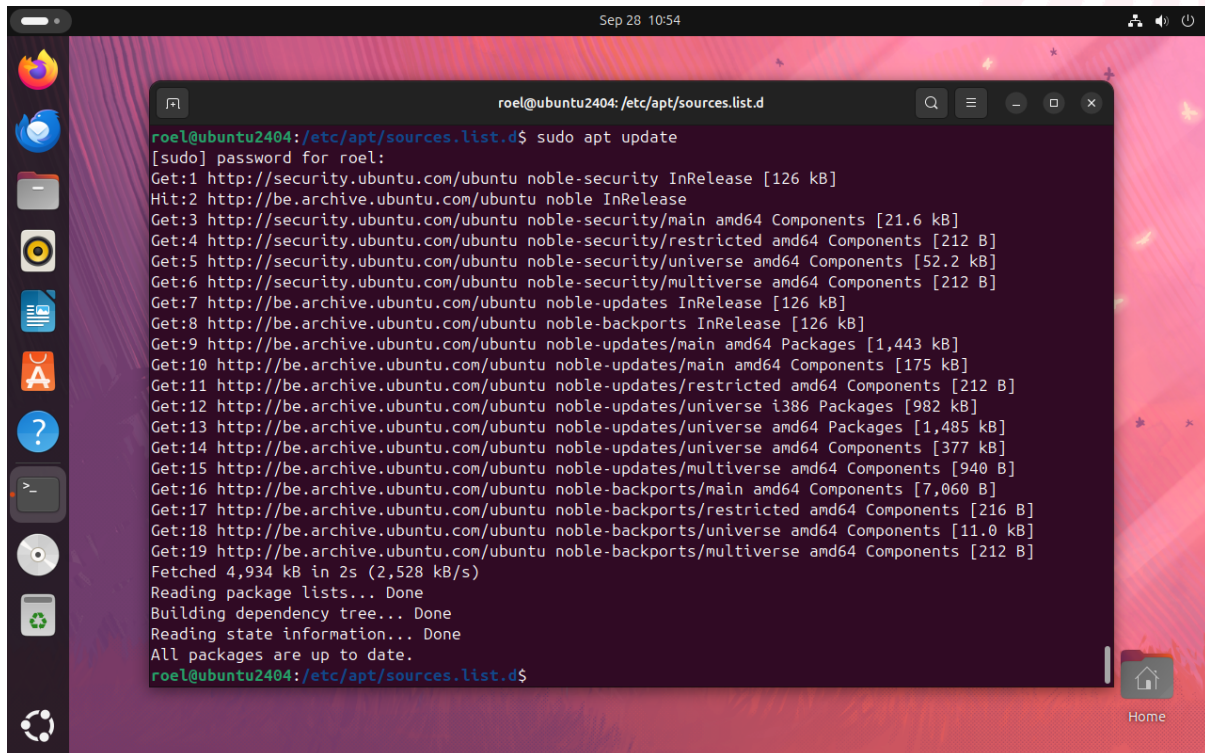
Een package manager haalt altijd zijn info van 1 of meerdere *repositories*. De *repositories* van Ubuntu worden bijgehouden in de map `/etc/apt/sources.list.d`. De standaard *repositories* staan in `/etc/apt/sources.list.d/ubuntu.sources`. Je kan met de inhoud van dit bestand zien met `cat`:

```
cat /etc/apt/sources.list.d/ubuntu.sources
```

Eventuele extra *repositories* komen in extra bestanden in deze map terecht.

5.10.2 Repositories updaten

We kunnen manueel aan de package manager vragen om al deze sources af te lopen en te kijken of er nieuwere versies van een pakket beschikbaar zijn. Dit doen we met het commando `sudo apt update`. De uitvoer zal er ongeveer zo uit zien:



```
roel@ubuntu2404: /etc/apt/sources.list.d
[roel@ubuntu2404: /etc/apt/sources.list.d$ sudo apt update
[sudo] password for roel:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://be.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:5 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:7 http://be.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:8 http://be.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:9 http://be.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,443 kB]
Get:10 http://be.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:11 http://be.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:12 http://be.archive.ubuntu.com/ubuntu noble-updates/universe i386 Packages [982 kB]
Get:13 http://be.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,485 kB]
Get:14 http://be.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:15 http://be.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:16 http://be.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,060 B]
Get:17 http://be.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:18 http://be.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.0 kB]
Get:19 http://be.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 4,934 kB in 2s (2,528 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
roel@ubuntu2404: /etc/apt/sources.list.d$
```

Iedere source (repository) die in onze sources stond wordt aangesproken, om te kijken of er nieuwe info is (nieuwe pakketten, updates van bestaande pakketten, ...) Wanneer een lijn met `Get` begint, wilt dit zeggen dat er nieuwe info was. Wanneer een lijn met `Hit` begint, wilt dit zeggen dat er geen nieuwe info was.

5.10.3 Software up-to-date houden

In tegenstelling tot wat het commando je doet vermoeden verandert `apt update` echter nog niks aan de software geïnstalleerd op ons systeem. `apt update` dient enkel en alleen om nieuwe info over de pakketten op te halen, niet om nieuwe versies van software te installeren. Hiervoor hebben we het commando `sudo apt upgrade`. Hiermee worden alle pakketten die op dit moment geïnstalleerd zijn automatisch geüpdatet naar de nieuwste versie.

5.10.4 Extra software installeren

APT kan je niet gebruiken om enkel bestaande software te updaten, maar je kan natuurlijk ook extra software bij installeren. Daarvoor gebruik je `apt install`. Probeer bijvoorbeeld het pakket `nginx` te installeren met:


```
sudo apt install nginx
```

Als `apt` vraagt “Do you want to continue? [Y/n]” gebruik je ENTER om te bevestigen.⁷

Je zal het volgende als output krijgen:

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer
required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 564 kB of archives.
After this operation, 1,596 kB of additional disk space will be
used.
Do you want to continue? [Y/n]
Get:1 http://be.archive.ubuntu.com/ubuntu noble-updates/main amd64
nginx-common all 1.24.0-2ubuntu7.5 [43.4 kB]
```

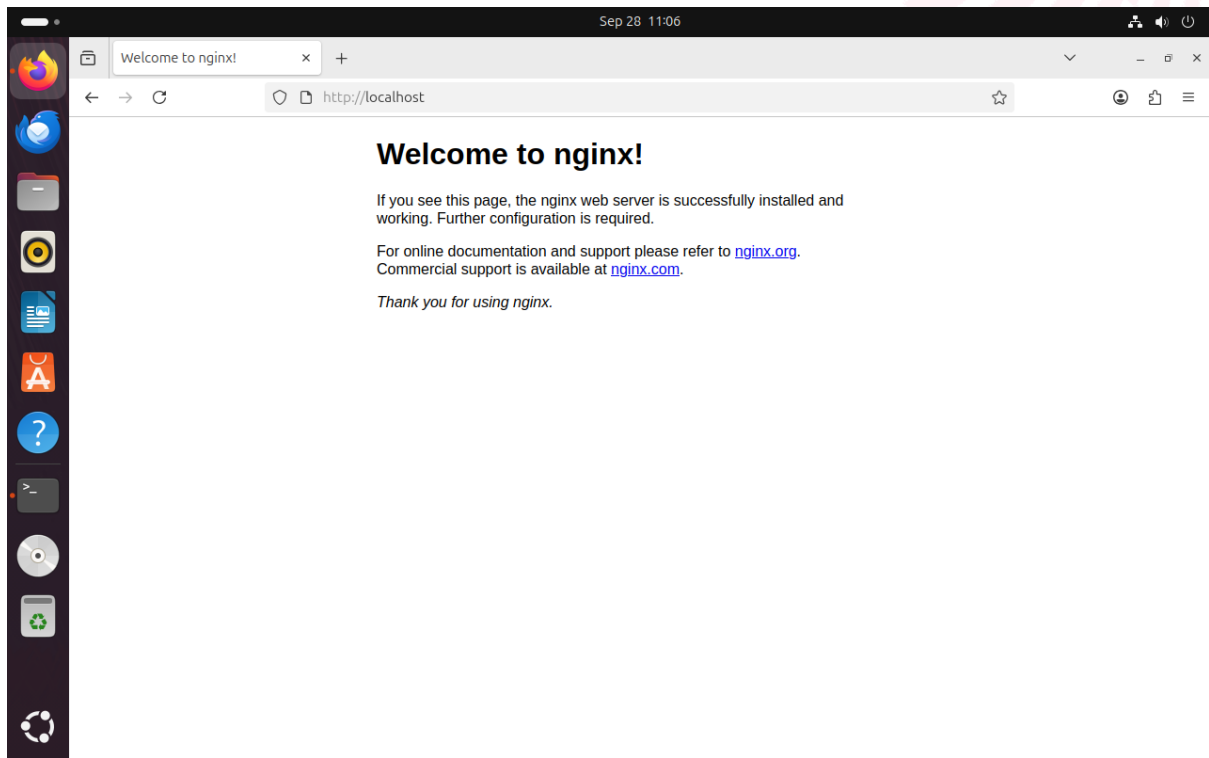
⁷ [Y/n] wil zeggen: ja (yes) of nee (no). Je ziet dat voor Y een hoofdletter wordt gebruikt: dit om aan te geven dat dit de default waarde is. Vandaar dat ENTER genoeg is om te bevestigen. Je kan ook explicieter bevestigen door y te typen gevolgd door ENTER. Je kan nee antwoorden met een n gevolgd door ENTER of gewoon het commando onderbreken met CTRL+C.



```
Get:2 http://be.archive.ubuntu.com/ubuntu noble-updates/main amd64
nginx amd64 1.24.0-2ubuntu7.5 [520 kB]
Fetched 564 kB in 1s (658 kB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 174111 files and directories currently
installed.)
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7.5_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7.5) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7.5_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7.5) ...
Setting up nginx-common (1.24.0-2ubuntu7.5) ...
Created symlink /etc/systemd/system/multi-
user.target.wants/nginx.service → /usr/lib/systemd/system/n
ginx.service.
Setting up nginx (1.24.0-2ubuntu7.5) ...
  * Upgrading binary nginx
[ OK ]
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for ufw (0.36.2-6) ...
```

Het eerste wat opvalt aan deze output is dat niet één maar meerdere pakketten (in dit geval 2) worden geïnstalleerd. Dit is zo omdat het pakket nginx een aantal *dependencies* heeft: pakketten die het nodig heeft om te kunnen werken. De package manager weet wat deze dependencies zijn en zal ze automatisch installeren.

Om wat context te geven: nginx is een webserver. Open wanneer je deze installatie gedaan hebt eens een browser en surf naar `http://localhost` (in je VM uiteraard). Je zal de standaardwebsite van nginx zien verschijnen:



Probeer nu eens met een teksteditor (zoals nano) het HTML-bestand in `/var/www/html` aan te passen en vernieuw daarna de webpagina in je browser.



Als gewone gebruiker heb je niet de juiste rechten om dit bestand aan te passen, dus zal je `sudo` voor je commando moeten zetten om dit als administrator te doen.

5.10.5 Software verwijderen

Als je nginx beu bent kan je dit eenvoudig de-installeren met:

```
sudo apt remove nginx
```

5.11 Rechten

Niet elke gebruiker mag zomaar elk bestand zien, aanpassen of uitvoeren. Stel dat gebruiker roel bijvoorbeeld gevoelige informatie in zijn home directory heeft staan, dan zouden andere gebruikers de inhoud van dat bestand niet mogen zien. Om die reden worden er in het bestandssysteem rechten bijgehouden voor elk bestand. Hier wordt het onderscheid gemaakt tussen rechten voor de:

1. **Owner:** de eigenaar van het bestand
2. **Group:** gebruikers die behoren tot de “groep” waartoe het bestand behoort
3. **Others:** alle andere gebruikers

Per categorie (owner, group, others) worden drie verschillende rechten toegekend:

1. **read:** leesrechten
2. **write:** schrijfrechten
3. **execute:** uitvoeringsrechten. Bij een gewoon bestand geeft dit aan dat dit een uitvoerbaar bestand is. Bij een directory heeft dit een andere betekenis (zie later).

5.11.1 Rechten bekijken

Je kan deze rechten zien met `ls -l` (-l staat voor “long listing”). De output kan er bijvoorbeeld zo uitzien:

```
total 4
drwxr-xr-x 2 roel roel 4096 Oct 30 13:21 dir
-rw-r--r-- 1 roel roel    0 Oct 30 13:20 file1
-r----- 1 roel roel    0 Oct 30 13:20 file2
-rwxr-x--- 1 roel role    0 Oct 30 13:20 file3
```

De laatste kolom geeft de naam van het bestand of map weer (dir, file1, file2, file3). We zijn vooral geïnteresseerd in kolom 1, 3 en 4.

Kolom 1 is 10 karakters lang en kan je verder opsplitsen in 4 groepjes:

- Eén karakter dat aangeeft over wat voor bestand het gaat. d geeft een directory aan, - geeft een normaal bestand aan.
- Drie groepjes van 3 karakters. Deze karakters zijn r, w, x of -.

Het zijn die groepjes van 3 karakters die de rechten aangeven. Het eerste groepje van 3 karakters zijn de rechten voor de owner, het tweede groepje de rechten voor de group en het laatste groepje de rechten voor others. **r** staat voor leesrechten (**r**ead), **w** voor schrijfrechten (**w**rite) en **x** staat voor uitvoeringsrechten (**e**xecute). Als er een letter staat wil dat zeggen dat een bepaalde groep van gebruikers dat recht heeft, als er een - staat wil dat zeggen dat die gebruikers dat recht niet hebben.

In onze voorbeeldoutput wil dat zeggen dat:

- `dir` een directory is (`d`), die leesbaar, schrijfbaar en “uitvoerbaar” is door de owner (`rwX`), leesbaar en “uitvoerbaar” door de group (`r-X`) en leesbaar en “uitvoerbaar” door others (`r-X`).
- `file1` een gewone file is (`-`), die leesbaar en schrijfbaar is door de owner (`rw-`), leesbaar door de group (`r--`) en leesbaar door others (`r--`).
- `file2` een gewone file is (`-`), die enkel leesbaar is door de owner (`r--`), en waar de group en others geen rechten op hebben (tweemaal `--`).
- `file3` een gewone file is (`-`), die leesbaar, schrijfbaar en uitvoerbaar is door de owner (`rwX`), leesbaar en uitvoerbaar door de group (`r-X`) en waar others geen rechten op hebben (`---`).

Wie zijn dan de owner en de group? Dat is wat kolom 3 en 4 aangeven. Alle bestanden in het voorbeeld hebben als owner “roel” en als group “roel”.



Een gebruiker kan tot één of meerdere groepen behoren. Wil je te weten komen tot welke groepen jouw gebruiker behoort? Gebruik dan het commando `id`. Dit zal je jouw `uid` geven (jouw user id met de naam tussen haakjes), jouw `gid` (de group id en naam van de hoofdgroep waar je toe behoort), en een lijst van alle groepen waar je toe behoort (met group id en naam).

5.11.2 Octale voorstelling van rechten

De rechten van een bestand worden opgeslagen als een octaal (8-tallig) getal met 3 cijfers. Het meest linkse cijfer geeft daarbij de rechten van de owner aan, het middelste cijfer de rechten van de group, en het meest rechtse cijfer de rechten van others.

Elk octaal cijfer kan je lezen als 3 bits: een read-bit, een write-bit en een execute-bit, zoals je kan zien in de volgende tabel:

Binair	Octaal	Rechten
100	4	Leesrechten
010	2	Schrijfrechten
001	1	Execute-rechten

Je kan deze combineren: 4 (100, leesrechten) en 1 (001, execute-rechten) geeft samen 5 (101, lees- en execute-rechten).

Als de rechten op bestand bijvoorbeeld worden voorgesteld als 755, dan wil dat zeggen:

- 7 voor de owner: lees-, schrijf-, en execute-rechten
- 5 voor de group: lees- en execute-rechten
- 5 voor others: lees- en execute-rechten

5.11.3 Rechten aanpassen

De rechten op een bestand aanpassen kan je doen zolang je de owner bent van dat bestand met het commando `chmod` (*change mode*).

Om bijvoorbeeld 755-rechten toe te kennen aan een bestand genaamd `file`, doe je:

```
chmod 755 file
```

Oefeningen

Probeer om het toekennen van rechten wat te oefenen de volgende oefeningen te maken (oplossingen op de volgende pagina):



Oefening 1: Maak een bestand met de rechten 764.



Oefening 2: Maak een bestand met de rechten `rwxr-r-x`



Oefening 3: Maak een bestand dat leesbaar en schrijfbaar is door de group, leesbaar en uitvoerbaar door de owner en leesbaar voor others.



Oefening 4: Maak een bestand aan dat enkel lees- en schrijfbaar is door de owner.

Oplossingen

Oefening 1

Dit is een redelijk straight forward gebruik van chmod:

```
touch file1
chmod 764 file1
```

Dit vertaalt naar lees-, schrijf- en uitvoerrechten voor de owner, lees- en schrijfrechten voor de group en leesrechten voor others.

Oefening 2

Hier zijn de rechten voorgesteld in de vorm die je zou krijgen als je `ls -l` uitvoert, `rwx` vertaalt naar 7 voor de owner, `r--` geeft 4 voor de group en `r-x` vertaalt naar 5 voor others:

```
touch file2
chmod 745 file2
```

Oefening 3

Let hier goed op! De rechten zijn hier van volgorde gewisseld: we geven eerst de rechten voor de group en vermelden daarna pas de rechten van de owner!

Leesbaar en schrijfbaar voor de group geeft ons een 6, lees- en uitvoerbaar geeft 5 en leesbaar voor others geeft een 4. Als we deze in de juiste volgorde zetten (user, group, others) krijgen we 564:

```
touch file3
chmod 564 file3
```

Oefening 4

Leesbaar en schrijfbaar door de owner geeft ons een 6. Alle andere rechten blijven op 0 staan:

```
touch file4
chmod 600 file4
```

5.11.4 Execute-rechten op een map

Bij een normaal bestand willen execute-rechten zeggen dat dit bestand uitvoerbaar is (je kan het dus runnen als een commando). Als je bijvoorbeeld een bestand genaamd mijn-programma wil runnen kan je ./mijn-programma gebruiken. De ./ zorgt ervoor dat het programma niet opgezocht in de PATH-variabele, maar dat het gewoon vanuit de huidige map wordt gestart.

Bij een directory is het minder duidelijk wat “uitvoerbaar” eigenlijk betekent. Daar slaat execute-rechten op het feit dat de gebruiker toegang heeft tot de inhoud van een map. Een gebruiker met leesrechten op een map kan de lijst van bestanden in de map zien (dus bijvoorbeeld ls gebruiken). Een gebruiker met execute-rechten op een map kan aan de inhoud van de map.

Om dit wat beter te begrijpen kan je dit zelf eens proberen. De volgende commando's maken 2 mappen aan in de map /tmp/demo, met in elke map 1 bestand met de string “Hello” in:

```
mkdir /tmp/demo
cd /tmp/demo
mkdir dir1 dir2
echo Hello >dir1/file1
echo Hello >dir2/file2
```

Verander nu de rechten van dir1 naar 100 (execute-rechten voor de owner) en de rechten van dir2 naar 400 (leesrechten voor de owner):

```
chmod 100 dir1
chmod 400 dir2
```

Als je nu ls -l doet zou dat volgende output moeten geven:

```
total 8
d--x----- 2 roel roel 4096 Oct 30 15:12 dir1
dr----- 2 roel roel 4096 Oct 30 15:12 dir2
```

Probeer nu de volgende commando's:

```
ls -l dir1
ls -l dir2
cat dir1/file1
cat dir2/file2
```

Je zal nu het volgende merken:

1. Commando 1 geeft een permission denied: je hebt geen leesrechten op de map dus je kan de lijst van files niet zien.
2. Commando 2 toont wel dat `file2` bestaat, maar verder kunnen er geen details worden opgehaald van dit bestand.
3. Commando 3 toont de inhoud van `file1` zonder error: hoewel we geen lijst kunnen krijgen van alle bestanden die zich in `dir1` bevinden, kunnen we wel aan deze bestanden.
4. Commando 4 geeft "Permission denied".

In de praktijk wil je in de meeste gevallen zowel lees- als execute-rechten geven aan een map. Merk op dat een map automatisch execute rechten krijgt als je die aanmaakt met `mkdir`, terwijl een normaal bestand gemaakt met `touch` deze rechten niet krijgt.

5.11.5 De root-gebruiker

Op een Linux-systeem is er altijd één speciale gebruiker: de root-gebruiker met als user id 0. Dit is de gebruiker die gebruikt voor administratie van het systeem. Deze gebruiker heeft dus alle rechten en negeert de rechten die op een bestand staan.⁸

Om commando's uit te voeren als de root-gebruiker kan je het `sudo`-commando gebruiken. Zet dit commando voor eender welk ander commando om het als root uit te voeren. Je hebt dit bijvoorbeeld al gedaan om packages te installeren met `apt`.

⁸ Execute-rechten zijn een special geval: de root-gebruiker zal nog steeds weigeren om een bestand uit te voeren zonder execute-rechten. Dit om te vermijden dat bestanden die niet bedoeld zijn om uitgevoerd te worden toch per ongeluk uitgevoerd worden. De root-gebruiker mag natuurlijk altijd wel `chmod` uitvoeren om dir rechten te kunnen geven.

Test het eens zelf uit: maak een bestand aan, zet de rechten op 000 en wis het daarna als root. Je zal merken dat dit zonder problemen lukt:

```
touch file  
chmod 000 file  
sudo rm file
```

5.12 Zoeken

Om bestanden te doorzoeken in de command line kan je het grep-commando gebruiken. Dat gebruik je als volgt:

```
grep zoekterm bestand
```

Als we bijvoorbeeld willen zoeken op “student” in het bestand /etc/passwd:

```
grep student /etc/passwd
```

Oefeningen

Probeer zelf op basis van de informatie die de manpage van grep geeft de volgende commando's te schrijven (de oplossingen staan op de volgende pagina):



Oefening 1: Schrijf een commando om alle lijnen in het bestand /etc/ssh/ssh_config weer te geven die eindigen op “no”. Tip: zoek op “Anchoring” in de manpage van grep.



Oefening 2: Schrijf een commando om alle lijnen in het bestand /etc/passwd weer te geven die beginnen met “root”.



Oefening 3: Schrijf een commando om alle lijnen in /etc/resolv.conf te tonen die **niet** zijn uitgecomment. Een uitgecommente lijn begint met #. Je zal hiervoor moeten zoeken hoe je met grep de lijnen moet vinden die de zoekterm **niet** matchen.

Oplossingen

grep kan gebruik maken van *reguliere expressies* om te zoeken. Eén feature van die reguliere expressies is “anchoring”: aangeven waar het begin en einde van de lijn moeten komen (met ^ en \$).

Oefening 1

\$ geeft het einde van een lijn aan, je wil dus zoeken op de zoekterm “no” gevolgd door het einde van de lijn:

```
grep no$ /etc/ssh/ssh_config
```

Oefening 2

^ geeft het begin van een lijn aan, je wil dus zoeken op het begin van de lijn, gevolgd door de zoekterm “root”:

```
grep ^root /etc/passwd
```

Oefening 3

Hiervoor heb je de optie --invert-match (of verkort -v) nodig. In de manpage lees je dit:

```
-v, --invert-match  
    Invert the sense of matching, to select non-matching lines.
```

We willen zoeken op de lijnen die niet beginnen met een hekje (#), dus we willen zoeken op het begin van de lijn (^) gevolgd door #, en dit dan inverteren:

```
grep -v ^# /etc/resolv.conf
```


5.13 Pipes

Tot nu toe hebben we enkel een commando op zich gebruikt. De echte kracht van de command line begint echter tevoorschijn te komen als we commando's gaan combineren om meer complexe bewerkingen te doen.

Eén zo'n manier om commando's te combineren is de pipe (`|`). Een pipe neemt de output van het commando links van de pipe en geeft dat als input aan het commando rechts van de pipe. Je kan dit voorstellen als een buis waardoor de output van het ene commando naar de input van het andere commando stroomt. We kunnen bijvoorbeeld met het volgende commando zoeken naar de string "group" in de output van `ls -l /etc`:

```
ls -l /etc | grep group
```

We gebruiken hier het `grep`-commando zonder bestand. In dat geval zal `grep` zoeken in zijn input. We nemen hier dus de output van `ls -l /etc` en geven dat door aan het commando `grep group`.

Je zou dit ook kunnen gebruiken om meerdere zoekopdrachten te combineren:

```
cat /usr/share/dict/american-english | grep ^f | grep s$
```

Apart doen deze commando's het volgende:

- `cat /usr/share/dict/american-english` output de inhoud van het bestand `/usr/share/dict/american-english`.
- `grep ^f` zoekt naar een "f" op het begin van de lijn.
- `grep s$` zoekt naar een "s" op het einde van de lijn.

Gecombineerd neemt dit dus de inhoud van `/usr/share/dict/american-english`, filtert dit op alle lijnen die beginnen met een "f" en filtert het daarna nog eens op alle lijnen die beginnen met een "s". Het eindresultaat is dat we alle lijnen in `/usr/share/dict/american-english` zien die beginnen met een "f" en eindigen op een "s".

Oefeningen

Probeer de volgende oefeningen nu eens te maken (oplossingen op de volgende pagina):



Oefening 1: Schrijf een commando dat alle gewone bestanden oplijst in /etc die leesbaar, schrijfbaar en uitvoerbaar zijn door de owner. Gebruik hiervoor de commando's ls en grep.



Oefening 2: Schrijf een commando dat alle lijnen output in /usr/share/dict/american-english die beginnen met "li" en eindigen op "se".



Oefening 3: Schrijf een commando dat alle **gewone bestanden** oplijst in /var/log die schrijfbaar zijn door de group. Hiervoor zal je moeten uitzoeken hoe je een wildcard "any" karakter doorgeeft in jouw zoekterm.

Oplossingen

Oefening 1

In sectie 5.8 hebben we geleerd dat de output van `ls -l` in de eerste kolom het soort bestand en de rechten erop output. We zoeken op alle **gewone** bestanden, dus we willen matchen op lijnen die beginnen met `-`. We willen lees-, schrijf- en uitvoeringsrechten voor de owner, dus de `-` moet gevolgd worden door `rwX`. We willen ook **alle** bestanden. Dit wil zeggen dat we ook verborgen bestanden willen zien (dit zijn bestanden die beginnen met een punt (`.`)). We moeten dus aan `ls -l` ook de optie `-a` geven. Het resultaat is:

```
ls -la /etc | grep ^-rwx
```

Oefening 2

Deze oefening is zeer gelijkaardig aan het voorbeeld dat we eerder hebben gezien:

```
cat /usr/share/dict/american-english | grep ^li | grep se$
```

Oefening 3

De wildcard character voor `grep` is de punt (`.`). In sectie 5.8 hebben we geleerd dat de rechten van de group het middelste groepje van de rechten is in de eerste kolom van de output van `ls -l`. We willen dus de rechten van de gebruiker overslaan. Het maakt ons niet uit of de group leesrechten heeft, we zoeken enkel naar de schrijfrechten. Dit geeft als commando:

```
ls -la /var/log | grep ^....w
```

5.14 Output redirection

Wat als we de output van een commando willen wegschrijven naar een bestand? Hier kunnen we output redirection voor gebruiken. Daarvoor gebruiken we het groter-dan teken (>). Je kan hierin een pijl zien die wijst naar het bestand waar de output in terecht moet komen.

Om bijvoorbeeld de string "Hello world" in het bestand file1 te zetten kunnen we het volgende commando uitvoeren:

```
echo "Hello world" >file1
```

De enkele pijl (>) overschrijft altijd het bestand. Als je de inhoud van het bestand wil behouden en enkel iets op het einde van het bestand erbij wil schrijven kan je de dubbele pijl (>>) gebruiken:

```
echo "Hello world" >file2  
echo "Goodbye world" >>file2
```

Check nu met cat de inhoud van file2. Voer vervolgens het volgende commando uit:

```
echo "Hi" >file2
```

Je zal zien (check het met cat) dat de inhoud van file2 is overschreven.



Wees voorzichtig als je output redirection gebruikt: kijk goed na dat je een >> gebruikt als je wil appenden en niet per ongeluk een >.

5.15 Tekst transformeren

Er bestaan verschillende commando's om tekst te transformeren, die vooral handig worden als je ze gaat combineren met andere commando's met behulp van pipes (zie sectie 5.13). In deze sectie zullen we het hebben over twee zo'n commando's: `cut` en `tr`.

5.15.1 `cut`

Met het `cut`-commando kunnen we stukjes tekst "uitsnijden".

Neem bijvoorbeeld het bestand `/etc/lsb-release`. Als je dat bestand bekijkt⁹ zal je zien dat dit bestaat uit lijnen met een variabelenaam en een waarde, gescheiden door een gelijkheidsteken (=). Als we enkel de waarden uit dat bestand willen halen moeten we dit bestand gaan splitsen op =.

Dit kunnen we als volgt met `cut` doen:

```
cut -d = -f 2 /etc/lsb-release
```

Met de optie `-d` (**d**elimiter) kunnen we bepalen waar we op willen splitsen (= in dit geval) en met de optie `-f` (**f**ields) kunnen we bepalen welke kolom (veldje) we willen behouden (te tellen vanaf 1). In dit geval willen we de tweede kolom dus gebruiken we `-f 2`. We kunnen ook meerdere kolommen (of veldjes) eruit halen. Dit kunnen we doen door de kolomnummers te scheiden met een komma.

Oefening

Probeer de volgende oefening (de oplossing staat op de volgende pagina):



Het bestand `/etc/passwd` bevat informatie over alle gebruikers van het systeem. Elke lijn bevat meerdere veldjes gescheiden door een `:`. Met welk commando kunnen we enkel de gebruikersnaam (eerste kolom) en de user id (derde kolom) voor elke gebruiker hieruit halen?

⁹ Met `cat /etc/lsb-release` uiteraard

Oplossing

We moeten splitsen op : en kolom 1 en 3 outputten¹⁰, dus het commando wordt:

```
cut -d : -f 1,3 /etc/passwd
```

5.15.2 tr

Een ander soort teksttransformatie die we kunnen doen is het vervangen van bepaalde karakters door andere karakters. Hiervoor kan je het commando `tr` (translate) gebruiken. In `/etc/passwd` wordt bijvoorbeeld ":" gebruikt om kolommen te scheiden. Als we dit liever met komma's doen kunnen we dat als volgt:

```
cat /etc/passwd | tr ':' ','
```

`tr` neemt als eerste argument de karakters die vervangen dienen te worden en als tweede argument de karakters waardoor deze vervangen moeten worden.

Oefening

Probeer de volgende oefening (de oplossing staat op de volgende pagina):



Schrijf een commando om in `/usr/share/dict/american-english` alle klinkers (a, e, i, o en u) te vervangen door "*".

¹⁰ Door bijvoorbeeld `man 5 passwd` te raadplegen kunnen we te weten komen dat dit kolom 1 en 3 zijn.

Oplossing

```
cat /usr/share/dict/American_english | tr aeiou '*'
```

5.16 Gecombineerde oefening

We kunnen al het voorgaande combineren om complexere taken te doen zoals:



Toon alle gewone bestanden in /var/log met enkel hun permissies, bestandsgrootte en bestandsnaam.

We kunnen dit aanpakken door op te splitsen wat er gevraagd wordt. We weten hoe we kunnen tonen wat er in de map /var/log zit, dus dat geeft ons als eerste stap:

```
ls -la /var/log
```

We willen echter enkel alle “gewone bestanden”. We weten dat dat wordt aangegeven met een “-” op het begin van de lijn, dus we kunnen grep gebruiken om enkel deze te tonen:

```
ls -la /var/log | grep ^-
```

Nu bevat onze output nog te veel informatie. We willen enkel de permissies (kolom 1), de bestandsgrootte (kolom 5) en de bestandsnaam (kolom 9). We weten dat we daar cut voor kunnen gebruiken:

```
ls -la /var/log | grep ^- | cut -d ' ' -f 1,5,9
```

Nu botsen we echter op een probleem: dit geeft niet exact wat we verwachten. Dit komt omdat er *meerdere spaties* zijn tussen elke kolom in de output van `ls -la`.

We kennen een commando om karakters te vervangen, namelijk `tr`. Als we naar de manpage van `tr` gaan kijken, dan vinden we de optie `-s` of `--squeeze-repeats`. We kunnen dit gebruiken om herhalingen van de spatie te reduceren naar slechts één spatie:

```
ls -la /var/log | grep ^- | tr -s ' '
```

Nu kunnen we tenslotte terug `cut` gebruiken om de juiste kolommen hieruit te halen:

```
ls -la /var/log | grep ^- | tr -s ' ' | cut -d ' ' -f 1,5,9
```