

# Computersystemen en -architectuur

## Introductie UNIX

*1 Ba INF*  
*2022–2023*

Kasper Engelen  
kasper.engelen@uantwerpen.be

Deze les is bedoeld om zelfstandig enkele belangrijke commando's en concepten uit Unix te leren kennen en te leren hoe en waar je ze kan gebruiken. Bijna alle commando's die je nodig hebt om de opdrachten te maken worden eerst stap voor stap uitgelegd aan de hand van interactieve voorbeelden. Voor enkele opdrachten kan het nodig zijn dat je zelf wat opzoekwerk moet verrichten om een nuttig commando te vinden. Zelfstandig dingen kunnen opzoeken is ook een belangrijke skill voor een informaticus. Zit je vast dan kan je uiteraard om hulp vragen aan de assistent.

De bedoeling is dat je tijdens het lezen en leren ook commando's uitvoert in je eigen terminal om zo direct het effect van het commando te kunnen zien. Commando's in een grijs kader zijn degene die je zou moeten uitvoeren in je terminal:

```
commando argument1 argument2 ...
```

Typ deze over in je eigen terminal, op deze manier krijg je de verschillende commando's snel in de vingers. Voor sommige oefeningen staat het resultaat dat je moet krijgen erbij in een blauw kader:

```
resultaat 1  
resultaat 2  
...
```

Af en toe zal je ook opdrachten tegenkomen in de groene kaders. Hier is het de bedoeling dat je zelf nadenkt en een commando, of reeks van commando's bedenkt die tot het gevraagde resultaat komen.

### Voorbeeld Opdracht

Hier staan de instructies.

## 1 De terminal openen

Op Ubuntu kan je de terminal openen met de toetsencombinatie **ctrl + alt + t**. Op MacOS kan je dit doen door Launchpad te openen en te zoeken naar "Terminal".

## 2 Files en Directories

### 2.1 Navigeren in het filesystem

Onze shell (terminal) bevindt zich steeds in een bepaalde directory in het filesystem. Deze directory noemen we de *Current Working Directory* (de directory van waaruit we werken). Al de commando's die we uitvoeren in de terminal worden uitgevoerd in of vanuit deze map. Het is daarom dus belangrijk om altijd te weten wat de Current Working Directory is. Indien je het niet meer weet of het volledige pad (absolute pad) nodig hebt naar je huidige locatie dan kan je het volgende commando gebruiken:

```
pwd
```

```
/home/student
```

Nu we weten wat onze huidige directory is, kunnen we ook aan het systeem vragen om te laten zien welke files en folders er zich allemaal in deze locatie bevinden. Dit doen we door gebruik te maken van het *List Directory* commando:

```
ls
```

```
Desktop Documents Downloads Music Pictures ...
```

Wanneer we dit commando uitvoeren zonder extra parameters dan laat deze de inhoud van de huidige directory (de current working directory) zien. Willen we echter de inhoud van een andere directory zien zonder onze Current Working Directory aan te passen dan kunnen we de locatie van deze directory meegeven als parameter bij het commando. Als we willen ophoofden welke files er zich in de root (/) van ons systeem bevinden, schrijven we:

```
ls /
```

```
bin boot dev etc home lib opt usr ...
```

Elk commando kan verschillende parameters als input krijgen. Deze parameters hebben een invloed op hoe het commando moet uitgevoerd worden en op welke files, directories, processen, etc. Parameters worden vaak aangegeven door één of meerdere liggende streepjes (-). Zo krijgen we door de optie `-l` toe te voegen aan `ls` ook een overzicht van de lees- en schrijfrechten en het formaat van alle files. Let op dat `l` de kleiner letter `l` is en niet het cijfer `1`!

```
ls -l /
```

Om een overzicht te krijgen van het gebruik en mogelijke parameters van een bepaald commando kan je altijd het commando `man` gebruiken. Dit geeft de instructiepagina's van het opgegeven commando weer. Voor het `ls` commando krijgen we dan bijvoorbeeld:

```
man ls
```

Opmerking: om `man` te sluiten druk je op de letter `q`.

### Opdracht 1

Geef alle persoonlijke mappen weer van gebruikers op dit systeem. De persoonlijke mappen bevinden zich in `/home`.

Door middel van het *Change Directory* commando `cd` kunnen we navigeren doorheen het filesystem (onze working directory aanpassen). We doen dit door de gewenste nieuwe locatie mee te geven als parameter aan het commando.

```
cd ..
```

De `..` staat voor de parent-directory, het commando brengt ons dus één niveau “hoger” in het filesystem. Let op dat dit commando geen output weergeeft, enkel indien de gewenste locatie niet bestaat of niet toegankelijk is zal het commando dit rapporteren. Gebruik nu onderstaand commando om te controleren dat onze locatie inderdaad is gewijzigd:

```
pwd
```

```
/home
```

Het Change Directory commando kan ook gebruikt worden zonder parameter. In dit geval wordt de huidige directory gewijzigd naar de persoonlijke home directory van de user. Er zijn twee verschillende commando's om dit te doen:

```
cd
```

```
cd ~
```

De terminal ziet er nu normaal gezien zo uit:

```
student@pclab25:~$
```

Gebruik nu `pwd` om te controleren dat we daadwerkelijk terug in de home directory van de gebruiker zitten. Het Change Directory commando kan zowel gebruik maken van een absoluut pad als van een relatief pad.

Als we willen weten of we in de persoonlijke home directory of in de root zitten, kijken we naar de tekst net voor het `$` teken. Indien hier `/` staat zitten we in de root:

```
student@pclab25:/$
```

Indien hier een tilde (`~`) staat, zitten we in de persoonlijke home folder:

```
student@pclab25:~$
```

### Opdracht 2

Gebruik **cd** om vanuit je persoonlijke home folder naar de root folder te gaan (zorg er eerst dus voor dat je zeker in je home folder zit) door gebruik te maken van een **absoluut pad**.

### Opdracht 3

Gebruik nu **cd** om vanuit je persoonlijke home folder naar de root folder te gaan (zorg er eerst dus voor dat je zeker in je home folder zit) door gebruik te maken van een **relatief pad**.

### Opdracht 4

Ga nu vanuit de root folder terug naar je eigen home folder door gebruik te maken van een **relatief pad**.

Zorg ervoor dat je **pwd**, **ls** en **cd** goed onder de knie hebt, je zal deze in je verdere loopbaan (zowel op de universiteit als in het bedrijfsleven) nog heel vaak nodig hebben. Speel nog wat met deze commando's zodat ze in je vingers zitten en je niet meer hoeft na te denken over hun gebruik. Spendeer ook nog wat extra aandacht aan het begrijpen van absolute en relatieve paden.

## 2.2 Beheren van mappen

**Opmerking:** Op de computers in de klaslokalen mag je enkel mappen aanmaken en verwijderen in de persoonlijke home directory. Ga dus eerst naar de persoonlijke home folder:

```
cd ~
```

We hebben gezien hoe we kunnen navigeren in het filesysteem en de huidige files en directories kunnen weergeven. Nu gaan we verder met het aanmaken en verwijderen van deze directories. Om een nieuwe directory aan te maken gebruiken we het volgende commando:

```
mkdir new_directory
```

Als parameter geven we een bestaand pad samen met de naam van de nieuwe directory op. In het vorige voorbeeld willen we in de huidige directory de map **new\_directory** toevoegen. Wanneer we nu de inhoud van onze huidige directory weergeven, zien we onze nieuwe directory in de lijst staan.

```
ls
```

```
Desktop Documents Downloads Music new_directory Pictures ...
```

We kunnen nu navigeren naar deze nieuwe directory, wanneer we in de directory de inhoud weergeven zullen we zien dat deze directory leeg is.

```
cd new_directory
ls
```

Eerst keren we terug naar onze home folder.

```
cd
```

Om een lege directory te verwijderen hebben we een ander commando:

```
rmdir new_directory  
ls
```

`rmdir` werkt enkel op lege directories. Op deze manier verwijder je niet per ongeluk bestanden, maar moet je ze eerst expliciet zelf verwijderen.

### Opdracht 5

Maak een nieuwe mappenstructuur in je home folder die er als volgt uitziet:

- Maak een map **Nieuw** aan in de persoonlijke map van de gebruiker
- Deze nieuwe map bevat twee mappen: **Map1** en **Map2**
- **Map1** bevat op zijn beurt twee mappen: **1** en **2**
- **Map2** bevat één map: **1**

Verwijder nu de volledige directory **Map2**.

## 2.3 Zoeken in het filesystem

**Opmerking:** om aan dit deel te kunnen werken moeten je eerst Opdracht 5 hebben afgewerkt! We bevinden ons nu in de map **Nieuw**.

Net als andere systemen bevat Unix ook een commando om een file of directory te zoeken. Dit commando heet, heel toepasselijk, **find**. Voeren we dit commando uit zonder opties of parameters dan krijgen we een recursieve oplijsting van alle files en directories in de huidige directory. Dit wil zeggen dat ook de inhoud van elke directory wordt weergegeven, alsook de inhoud van directories in elke directory, etc.

```
find
```

De output is dan:

```
.  
./Map1  
./Map1/1  
./Map1/2
```

Uiteraard kunnen we ook meer in detail gaan zoeken. Zo kunnen we meegeven als parameter in welke directory of directories het systeem moet zoeken. Verder kunnen we opgeven welke naam de file die we zoeken heeft (of welk patroon) en welk type we zoeken (file **f** of directory **d**). Om te zoeken op naam moeten we expliciet de optie **-name** meegeven voor de naam van de file. Let op dat indien een filename een spatie bevat je de volledige filename tussen quotes moet zetten.

```
find ./ -name 'Map1'
```

```
find ./ -name 'Map*'
```

```
./Map1
```

Willen we zoeken op type dan geven we de optie `-type` mee.

```
find ./ -type d -name 'Map*'
```

```
./Map1
```

```
find ./ -type f -name 'Map*'
```

Merk op dat er in het tweede geval geen files worden gevonden. Als we nu een file `Map32` aanmaken, en opnieuw zoeken naar een file, krijgen we een ander resultaat:

```
touch Map32.txt  
find -type f -name 'Map*'
```

```
./Map32.txt
```

## Opdracht 6

Zoek alle directories binnen de directory `/etc` die beginnen met 's'.

## 2.4 Files weergeven

Ga eerst naar de persoonlijke home folder:

```
cd ~
```

Voor we verder gaan, introduceren we eerst een ander commando: `wget`, dit commando kan een file van het internet downloaden. Volgend commando downloadt een *tgz-archieff* van het internet en slaat dit op in de huidige directory.

```
wget http://msdl.uantwerpen.be/people/hv/teaching/ComputerSystemsArchitecture/materials/CS1/Names.tgz
```

Eerst moeten we dit *tgz-archieff* uitpakken om aan de files te kunnen. Doe dit door gebruik te maken van het volgende commando (waarbij `x` staat voor *extract*):

```
tar -xvzf Names.tgz
```

Als je nu het commando `ls` uitvoert zal je zien dat er in de huidige directory drie nieuwe files staan.

```
ls
```

De eerste is het tgz-archief **Names.tgz**, de andere twee zijn tekstfiles die in het archief zaten: **male-names** en **female-names**. Deze twee files zullen we verder gebruiken om te leren hoe we inhoud van een file kunnen weergeven en hoe we dit kunnen aanpassen. We kunnen de inhoud van een file weergeven door gebruik te maken van het volgende commando:

```
cat female-names
```

**cat** gebruiken we wanneer we een volledige file willen weergeven in onze terminal. Soms zijn we echter enkel geïnteresseerd in de bovenste lijnen van een file, hiervoor gebruiken we volgend commando:

```
head female-names
```

Standaard worden de eerste 10 lijnen van een file weergegeven. Willen we meer of minder lijnen weergeven, dan kunnen we dit aanpassen door gebruik te maken van de optie **-n** gevolgd door het aantal regels.

```
head -n 25 female-names
```

Als tegenhanger van **head** bestaat er een gelijkaardig commando op de laatste lijnen van een file te bekijken (vooral handig voor het inspecteren van log-files).

```
tail -n 20 female-names
```

Buiten het weergeven van files kunnen we ook zoeken naar woorden in een bepaalde file. Hiervoor gebruiken we het volgende commando om **tom** te zoeken in de lijst met mannen namen.

```
grep tom male-names
```

Merk op dat **grep** zoekt naar voorkomens van de drie karakters **t o m** (in deze volgorde) ook al worden deze karakters gevolgd of voorafgegaan door andere karakters.

### Opdracht 7

Zoek je eigen naam op in de lijst met namen.

We kunnen ook weergeven hoeveel lijnen, woorden en bytes een bepaalde file bevat. Hiervoor gebruiken we het *Word Count* commando:

```
wc male-names
```

```
3906 3927 26784 male-names
```

Met de opties **-w** en **-l** kunnen we opgeven dat we enkel het aantal woorden, respectievelijk lijnen willen zien. Ook een combinatie van beide is mogelijk.

```
wc -l male-names
```

```
3906 male-names
```

```
wc -w male-names
```

```
3927 male-names
```

```
wc -w -l male-names
```

```
3906 3927 male-names
```

## 2.5 Files manipuleren

Nu we weten hoe we bestaande files kunnen bekijken, gaan we zien hoe we files kunnen aanmaken/verplaatsen/kopiëren/verwijderen. Een file kopiëren doen we met behulp van het *Copy* commando:

```
cp male-names new-male-names  
ls
```

Waarbij we als eerste parameter opgeven welke file we willen kopiëren en als tweede parameter de nieuwe locatie en naam van de kopie. **cp** kan geen nieuwe directories aanmaken, het opgegeven pad moet dus bestaan. Met het commando **ls** zien we dat er nu een file **new-male-names** is aangemaakt. Een file kan ook verwijderd worden door gebruik te maken van het commando **rm**. Als parameter geef je de file die verwijderd moet worden op.

```
rm male-names  
ls
```

We zien dat de file **male-names** niet langer bestaat. Bestaande files kunnen ook verplaatst worden binnen het filesysteem. Om dit te illustreren zullen we eerst een nieuwe directory **Names** aanmaken. We verplaatsen de files **female-names** en **new-male-names** naar deze directory.

```
mkdir Names
```

```
mv female-names Names/female-names
```

```
mv new-male-names Names/new-male-names
```

```
ls Names
```

Een expliciet *Rename* commando bestaat niet in Unix. Een rename kan beschouwt worden als het verplaatsen van een file naar dezelfde locatie maar met een andere naam. In volgend voorbeeld zullen we het bestand met de namen van mannen hernoemen naar de originele naam.

```
mv Names/new-male-names Names/male-names
```



```
ls Names
```

```
female-names male-names
```

Gebruik nu onderstaand commando om er zeker van te zijn dat we ons terug in je eigen home folder bevinden.

```
cd
```

## 2.6 De inhoud van files aanpassen

Indien we een file willen aanmaken en er tekst aan willen toevoegen, kunnen we dit doen door de **touch** en **echo** commands te combineren:

```
touch hello_file.txt
```

We hebben nu een nieuwe file **hello\_file.txt** aangemaakt. Als je **cat hello\_file.txt** doet zal je zien dat deze file leeg is. Om deze file wat inhoud te geven doen we het volgende:

```
echo 'Hello world!' > hello_file.txt
```

Als we nu de file openen met **cat** dan zien we dat de inhoud is veranderd:

```
cat hello_file.txt
```

```
Hello world!
```

## Opdracht 8

Zorg dat je je bevindt in de persoonlijke home directory.

1. Creëer twee nieuwe directories: **color** en **shape**.
2. Navigeer naar de directory **color** en maak de files **red**, **green**, **blue**, **apple** en **square** aan. (hint: gebruik het commando **touch** om een nieuwe file aan te maken)
3. Toon alle files in de huidige directory.
4. Verwijder **apple** aangezien dit geen kleur is.
5. Verplaats de file **square** naar de directory **shape**.
6. Pas de overgebleven files aan zodanig dat ze hun hexadecimale kleurencode bevatten als inhoud (**FF0000** voor rood, **00FF00** voor groen, **0000FF** voor blauw).  
Hint: om de tekst “abc 123” in een file **output\_file** te duwen doe je **echo 'abc 123' > output\_file**
7. Maak een kopie van **blue** naar een nieuwe file **purple** en pas de inhoud van deze nieuwe file aan naar **800080**.
8. Hernoem **red** naar **yellow** en pas de inhoud van de file aan naar **FFFF00**.
9. Geef alle files in de directory weer.

## 2.7 Processen

Buiten files bestaat een systeem ook nog uit lopende processen. Elk process heeft een (niet noodzakelijk unieke) naam en een unieke process identifier *PID*. In deze sectie zullen we vooral gebruik maken van het process **sleep**, als we dit commando uitvoeren dan wordt er een process gestart dat een aantal seconden slaapt (wacht) en dan eindigt. Onderstaand commando zal een process starten dat 3 seconden wacht en dan sluit.

```
sleep 3
```

Belangrijk om op te merken is dat tijdens de uitvoer van het process onze terminal geen andere input toestaat. Het process dat juist gestart is, is verbonden aan je huidige terminal. Sluit je de terminal, dan zal ook het process afgesloten worden. Wil je het uitvoeren van een lopend process zelf beëindigen, dan kan je gebruik maken van de toetsencombinatie **CTRL+C**. Probeer dit nu zelf door eerst een **sleep** commando te starten met als parameter **1000**. Sluit dit process vervolgens af voor het zelf tot een einde gekomen is.

```
sleep 1000
```

Standaard worden processen in de foreground (voorggrond) uitgevoerd. Dit heeft als consequentie dat de terminal niet beschikbaar is om andere commando's in te geven. We kunnen processen echter ook op de achtergrond runnen zodat onze terminal vrij blijft voor nieuwe commando's. Een process dat in de background loopt is niet verbonden aan een specifieke terminal. Het runnen van een commando in de background doen we door **&** achter het commando te zetten.

```
sleep 100 &
```

De terminal zal ons de PID van het process tonen wanneer we dit in de achtergrond starten zodat we dit later kunnen terugvinden in ons systeem. Om alle lopende processen te inspecteren kunnen we gebruik maken van de commando's **top** en **htop**.

```
top
```

We kunnen lopende processen ook beëindigen met het volgende commando indien we de PID kennen:

```
kill process_id
```

Indien we enkel de naam kennen kunnen we volgend commando gebruiken:

```
pkill process_name
```

**pkill** zal alle processen die dezelfde naam hebben een interrupt sturen zodat deze zichzelf afsluiten. Het gebruik van **kill** is veiliger dan **pkill**, omdat we bij **kill** expliciet vermelden welk process we willen beëindigen, terwijl **pkill** alle processen met dezelfde naam beëindigd. Een process dat in de achtergrond loopt kunnen we ook altijd terug naar de voorgrond brengen met het commando **fg**.

## 2.8 I/O Redirects

Ga terug naar de persoonlijk home folder:

```
cd ~
```

Processen schrijven hun output naar de *Standard Output* en nemen input van de *Standard Input*. In onze gevallen is de Standard Output de terminal en de Standard Input het keyboard. Verder bestaat er ook nog de *Standard Error* die voornamelijk, zoals de naam doet vermoeden, error boodschappen zal bevatten. We kunnen echter kiezen voor alternatieve I/O kanalen. We kunnen de output van een process laten wegschrijven naar een file in plaats van naar de terminal. Dit doen we door gebruik te maken van **>**, hieronder schrijven we de output van **ls** weg naar de file **output\_ls.txt**. Vervolgens kunnen we **cat** gebruiken om de inhoud van deze file te bekijken.

```
ls > output_ls.txt  
cat output_ls.txt
```

Voer nu onderstaande commando's uit. Het eerste commando zal de output van **ls -l** wegschrijven naar de file **output\_ls.txt** en het tweede zal de inhoud van deze file opnieuw weergeven.

```
ls -l > output_ls.txt  
cat output_ls.txt
```

We merken op dat onze oorspronkelijke output van het eerste **ls** commando niet meer in de file staat. De redirect **>** overschrijft bestaande files altijd volledig. Indien we enkel output willen

toevoegen aan bestaande files kunnen we `>>` gebruiken.

```
ls -l >> output_ls.txt
cat output_ls.txt
```

We zien nu dat de oorspronkelijke output nog steeds in de file aanwezig is.

Omgekeerd kunnen we ook een file als input nemen voor een commando. Hiervoor beschouwen we het commando `sort` dat de input file die het krijgt zal sorteren en weergeven in de Standard Output, zoals hieronder weergegeven:

```
sort < Names/male-names
```

Een nuttig commando in combinatie met de redirects is het commando `echo`, dit print de gegeven input terug out naar de Standard Output.

```
echo Hello, World!
```

Indien we de output van `echo` willen opslaan in een file doen we:

```
echo 'Hello, World!' > output_file.txt
```

### Opdracht 9

1. Navigeer naar de map **Names** (deze heb je aangemaakt in section 2.5)
2. Geef de laatste 25 mannennamen weer.
3. Geef de eerste 30 vrouwenamen weer.
4. Tel het aantal lijnen in beide files.
5. Print alle namen die de letter **q** bevatten.
6. De naam **Raf** komt niet voor in de lijst. Voeg deze naam toe aan het einde van de lijst doormiddel van redirecting van output.
7. Voeg alle mannen- en vrouwenamen samen in een file **all-names**. Zorg ervoor dat deze nieuwe file opnieuw gesorteerd is.
8. Print het aantal namen dat meer dan 1 keer voorkomt in **all-names**. (Hint: gebruik `uniq`)
9. Maak een nieuwe file **unisex-names** aan die alle namen bevat die zowel voor vrouwen als mannen gebruikt kunnen worden.

Op het einde van deze les zou je de volgende commando's moeten kennen en kunnen gebruiken: `pwd`, `ls`, `cd`, `man`, `mkdir`, `rmdir`, `find`, `wget`, `tar`, `cat`, `head`, `tail`, `grep`, `wc`, `cp`, `mv`, `sleep`, `top`, `kill`, `pkill`, `sort`, `echo`, `&`.

Hieronder vind je wat praktische informatie over UNIX, dit zijn geen oefeningen.

## A File permissions

Gebruikers hebben een unieke username, en maken deel uit van één of meerdere groepen. Om gebruikers (en groepen) toegang te verlenen tot bepaalde files, bestaan er zogenaamde permissions. Deze zijn read (**r**), write (**w**), en execute (**x**). De permissions van een file kunnen worden aangepast d.m.v. het commando **chmod**. Bijvoorbeeld: vooraleer een file **script.sh** kan worden uitgevoerd, moet men eerst het commando **chmod +x ./script.sh** uitvoeren.

## B Archiving

Via de command line kan men ook archieven aanmaken. Een zogenaamde *tarball* is een archief met daarin enkele files. Voor het werken met tar bestaan er onderstaande commando's:

- Inpakken: **tar -cvf tarball.tar files**
- Uitpakken: **tar -xvf tarball.tar**
- Files ophijsten: **tar -tf tarball.tar**
- Files aanpassen: **tar -uf tarball.tar files**

*Gzip* is kort voor “GNU zip”, en voorziet compressie voor tar files. De volgende commando's zijn bedoeld voor het comprimeren en decomprimeren van zulke files:

- Comprimeren: **gzip tarball.tar**
- Decomprimeren: **gunzip tarball.tar.gz**

Wanneer men tar en gzip combineert krijgt men de bestandsextensie **tar.gz** of **tgz**. Indien men uitsluitend gebruik maakt van tar in combinatie met gzip, kan men onderstaande commands gebruiken als afkorting:

- Inpakken + comprimeren: **tar -cvzf tarball.tgz files**
- Decomprimeren + uitpakken: **tar -xvzf tarball.tgz**
- Files ophijsten: **tar -tzf tarball.tgz**

## C Standard in- en output

Processes op een UNIX machine hebben input en output. Gewoon zijn dit de standard input (toetsenbord) en standard output (terminal). Er is ook een standard error voor foutmeldingen. Indien men de output van een script wilt wegschrijven naar een file, kan men gebruikmaken van *redirection*. Indien men de output van het éne proces wilt gebruiken als de input voor een ander proces, kan men ook gebruikmaken van *pipes*. In table 1 staan enkele voorbeelden.

Commando	Uitleg
<code>ls -alrF &gt; listing.txt</code>	De output van <code>ls</code> opslaan in <code>listing.txt</code>
<code>sort &lt; listing.txt</code>	De inhoud van <code>listing.txt</code> sorteren.
<code>echo Hello &gt;&gt; listing.txt</code>	De string <code>hello</code> toevoegen aan het einde van <code>listing.txt</code> .
<code>echo hello &gt; /dev/null</code>	De output van <code>echo</code> negeren.
<code>who 2&gt; errors.txt</code>	De errors van <code>who</code> opslaan in <code>errors.txt</code>
<code>who &gt; names.txt</code> <code>sort &lt; names.txt</code>	De output van <code>who</code> opslaan in <code>names.txt</code> en vervolgens sorteren (zonder pipes).
<code>who   sort</code>	De output van <code>who</code> sorteren (met pipes).

Tabel 1: Enkele voorbeelden van standard I/O en pipes.