

Computersystemen en -architectuur

Oplossingen introductie UNIX

*1 Ba INF
2022–2023*

Kasper Engelen
kasper.engelen@uantwerpen.be

In dit document vind je de oplossingen van de oefeningen over de UNIX shell. De informatie in dit document komt overeen met de computers in de lokalen M.G.025 en M.G.026. Indien je de oefeningen hebt gemaakt op je eigen computer kan het resultaat er iets anders uit zien.

De terminal commands staan in het grijs:

```
command argument1 argument2 ...
```

Merk op dat soms voor de duidelijkheid de volledige terminal tekst wordt vermeld. Bijvoorbeeld:

```
student@pclab25:/usr/include$ ls ./
```

De output en resultaten staan telkens in het blauw:

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

1 Opdracht 1

Ga naar de persoonlijke home folder `/home/student` met behulp van het `cd ~` commando. Om de inhoud van de map `/home` te zien, hebben we twee opties:

- We gebruiken een relatief pad:

```
student@pclab25:~$ ls ../
```

```
student sysman
```

- We gebruiken een absoluut pad:

```
student@pclab25:~$ ls /home
```

```
student sysman
```

Het `ls` commando geeft dus weer dat er in `/home` twee mappen `student` en `sysman` zitten.

2 Opdracht 2

We gaan naar onze persoonlijke home folder met het `cd ~` commando. De bevinden ons nu in `/home/student`. Als we nu met een absoluut pad naar de root willen gaan doen we `cd /`. Dit ziet er als volgt uit:

```
student@pclab25:~$ cd /
```

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

Dit pad begint met `/` en is dus een absoluut pad.

3 Opdracht 3

We bevinden ons wederom in `/home/student`. We gaan nu naar de root met behulp van een relatief pad door `cd ../../` te doen. Dit brengt ons twee niveaus omhoog. Ter illustratie:

```
student@pclab25:~$ cd ../../
```

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

Omdat dit pad begint met `..` en niet met een `/` spreken we van een relatief pad.

4 Opdracht 4

We bevinden ons in de root folder. Als we nu naar de gebruikersmap `/home/student` willen gaan met een relatief pad, kan dat als volgt:

```
student@pclab25:/$ cd ./home/student
```

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

Omdat dit pad begint met `./` en niet met een `/` spreken we van een relatief pad.

5 Opdracht 5

We bevinden ons in de persoonlijke home folder (`/home/student`).

1. Om een map `Nieuw` te maken doen we

```
student@pclab25:~$ mkdir Nieuw
```

2. We openen deze map met het `cd` commando:

```
student@pclab25:~$ cd ./Nieuw
```

3. We maken nu twee mappen `Map1` en `Map2`

```
student@pclab25:~/Nieuw$ mkdir Map1
```

```
student@pclab25:~/Nieuw$ mkdir Map2
```

4. Ter controle checken we met `ls` even of dit klopt:

```
student@pclab25:~/Nieuw$ ls ./
```

```
Map1 Map2
```

We zien dat er nu inderdaad twee mappen `Map1` en `Map2` zijn aangemaakt.

5. We gaan naar `Map1` en maken twee mappen 1 en 2 aan:

```
student@pclab25:~/Nieuw$ cd ./Map1
```

```
student@pclab25:~/Nieuw/Map1$ mkdir 1 2
```

6. Vervolgens gaan we naar `Map2` met een relatief pad:

```
student@pclab25:~/Nieuw/Map1$ cd ../Map2
```

7. We maken nu een map 1 aan:

```
student@pclab25:~/Nieuw/Map2$ mkdir 1
```

8. We gaan terug naar de map `Nieuw`:

```
student@pclab25:~/Nieuw/Map2$ cd ..
```

9. We verwijderen mappen 1 en `Map2`

```
student@pclab25:~/Nieuw$ rmdir ./Map2/1
```

```
student@pclab25:~/Nieuw$ rmdir ./Map2
```

10. Als we `ls` doen zien we dat enkel `Map1` overblijft:

```
student@pclab25:~/Nieuw$ ls
```

Map1

6 Opdracht 6

Omdat `/etc` een absoluut pad is, maakt het niet uit in welke map we ons momenteel bevinden. Concreet willen we hier alle directories (`-type d`) vinden die beginnen met de letter `s` (`-name 's*'`) en die zich bevinden in de map `/etc`. Als we dit alles combineren krijgen we het onderstaande commando:

```
find /etc -type d -name 's*'
```

Als je dit intypt in de terminal krijg je een lange lijst met directories te zien.

7 Opdracht 7

Voor deze opdracht pas je één van de onderstaande commando's toe. Vervang `naam` door je eigen naam.

```
student@pclab25:~$ grep 'naam' male-names
```

```
student@pclab25:~$ grep 'naam' female-names
```

8 Opdracht 8

1. We maken de twee mappen aan met `mkdir`:

```
student@pclab25:~$ mkdir color shape
```

2. We gaan naar de `color` directory en maken alle noodzakelijke files aan:

```
student@pclab25:~$ cd color
student@pclab25:~/color$ touch red green blue apple square
```

3. We tonen alle files in de huidige directory:

```
student@pclab25:~/color$ ls
```

```
apple blue green red square
```

4. We verwijderen de file `apple`:

```
student@pclab25:~/color$ rm apple
```

5. We verplaatsen de file `square` naar `shape` met een relatief pad:

```
student@pclab25:~/color$ mv ./square ../shape/square
```

6. We passen de inhoud van de files aan:

```
student@pclab25:~/color$ echo FF0000 > red
student@pclab25:~/color$ echo 00FF00 > green
student@pclab25:~/color$ echo 0000FF > blue
```

Indien we nu bijvoorbeeld de file `red` openen, zien we dat de inhoud is aangepast:

```
student@pclab25:~/color$ cat red
```

```
FF0000
```

7. We maken een kopie met `cp` en passen de inhoud van de file aan:

```
student@pclab25:~/color$ cp blue purple
student@pclab25:~/color$ echo 800080 > purple
```

Indien we de file `purple` openen zien we dat de inhoud is aangepast:

```
student@pclab25:~/color$ cat purple
```

```
800080
```

8. We doen ongeveer hetzelfde voor `red` en `yellow`:

```
student@pclab25:~/color$ cp red yellow
student@pclab25:~/color$ echo FFFF00 > yellow
```

9. Als we alle files willen weergeven gebruiken we `ls`:

```
student@pclab25:~/color$ ls ./
```

```
blue green purple red yellow
```

9 Opdracht 9

Voor deze opdracht moet je eerst de oefening van sectie 2.5 maken, zodat er in de persoonlijke home folder een map `Names` is met daarin twee files `female-names` en `male-names`.

1. We openen de map `Names` in de persoonlijke home folder:

```
student@pclab25:~$ cd Names
```

2. We gebruiken het **head** commando om de eerste lijnen te vinden:

```
student@pclab25:~/Names$ head -n 25 male-names
```

3. We gebruiken opnieuw het **head** commando:

```
student@pclab25:~/Names$ head -n 30 female-names
```

4. Met het **wc** commando kan men de lijnen in meerdere files tegelijk tellen:

```
student@pclab25:~/Names$ wc -l male-names female-names
```

```
3906 male-names
4955 female-names
8861 total
```

5. Om alle namen te vinden die een **q** bevatten gebruiken we het **grep** commando samen met het patroon **.*q.***. Merk op dat **.*** overeenkomt met elk mogelijk stuk tekst. Het patroon komt dus overeen met eender welk stuk tekst, gevolgd door een **q**, gevolgd door eender welk ander stuk tekst:

```
student@pclab25:~/Names$ grep '.*q.*' female-names male-names
```

6. We gebruiken **echo** om de naam naar de output te sturen en gebruiken **>>** om deze toe te voegen aan het eind van de file **male-names**:

```
student@pclab25:~/Names$ echo Raf >> male-names
```

7. Merk op dat **cat** meerdere files tegelijk kan weergeven. De output van **cat** redirecten we naar het **sort** commando. De output van **sort** wordt op zijn beurt geredirect naar de file **all-names**:

```
student@pclab25:~/Names$ cat male-names female-names | sort >
all-names
```

8. Om duplicaten te vinden gebruiken we **uniq -D**. Let wel op dat dit commando enkel werkt op gesorteerde files. Vervolgens halen we de duplicaten eruit met **uniq**. Tenslotte redirecten we alles naar **wc -l** om het aantal namen te tellen:

```
student@pclab25:~/Names$ uniq -D all-names | uniq | wc -l
```

```
419
```

9. Het idee hier is dat we **male-names** en **female-names** samenvoegen in één file. Alle namen die dan tweemaal voorkomen behoren tot beide files en moeten dus opgeslagen worden in **unisex-names**.

- (a) Verwijder de duplicaten in `male-names` en `female-names` afzonderlijk met `uniq`:

```
student@pclab25:~/Names$ uniq male-names > unique-male
student@pclab25:~/Names$ uniq female-names > unique-female
```

- (b) Voeg beide bestanden samen tot één gesorteerde file:

```
student@pclab25:~/Names$ cat unique-male unique-female | sort >
merged-sorted
```

- (c) Zoek nu alle duplicaten met `uniq -D` en sla telkens één van de duplicaten op (met `uniq`) in `unisex-names`:

```
student@pclab25:~/Names$ uniq -D merged-sorted | uniq >
unisex-names
```