

## Opdrachten III

### Opdracht 1. Hulp functie

De ingebouwde hulp functie van UNIX wordt met het `man` commando (voor *manual pages*) opgeroepen en geeft informatie over het aangegeven commando. Bijvoorbeeld geeft `man cp` informatie over het kopiëren van bestanden.

- Achterhaal wat het commando `du` doet. Pas dit commando in je home directory toe en interpreteer de uitvoer.
- Probeer het `du` commando ook met de `-k` en `-s` opties, nadat je hebt gevonden wat deze betekenen.
- Vind de juiste opties voor `ls` zo dat het commando de bestanden met uitvoerige informatie aangeeft, gesorteerd op de tijd wanneer de laatste wijziging heeft plaats gevonden, beginnend met de oudste.

### Opdracht 2. Invoer/uitvoer omleiding

Je kunt soms slimme dingen doen, door een paar commando's te combineren. Hiervoor is het handig als je in- en uitvoer kunt omleiden en dit gebeurt met `<`, `|` (pipe) en `>`.

- `cmd < file`: de invoer van het commando `cmd` komt uit het bestand `file` in plaats van het toetsenbord.
- `cmd > file`: de uitvoer van `cmd` gaat naar het bestand `file` in plaats van je scherm. Bijvoorbeeld sla je met `du -k ~ > du.out` de uitvoer van het `du` commando in het bestand `du.out` op, waar je het rustig kunt bekijken. (De `~` staat voor je home directory, hiervoor kan je ook de shell variabele `$HOME` gebruiken.)
- `cmd1 | cmd2`: de uitvoer van `cmd1` wordt de invoer van `cmd2`. Bijvoorbeeld sorteer je met `du -k ~ | sort -n` de uitvoer van het `du` commando op de grootte van de directories (ga de opties van `sort` zelf na).
- Geef van een tekst bestand (bijvoorbeeld `Grondwet.txt`) de regels 10 t/m 20 uit, door de `head` en `tail` commando's (met de goede opties) met een pipe `|` te combineren.

### Opdracht 3. Vinden van woorden en bestanden

- Met de commando's `grep` en zijn varianten `fgrep` en `egrep` kan men naar strings in een bestand zoeken. Gewoon geeft het commando de regels uit die de string bevatten. De string mag een *reguliere uitdrukking* zijn, bijvoorbeeld geeft `ls -l | grep '^d'` alle subdirectories van het actuele directory uit. Let op dat reguliere uitdrukkingen in *quotes* ingesloten moeten zijn (bijvoorbeeld `'^d'`).
- Vind in je home directory alle bestanden die *uitvoerbaar* zijn.
- Om naar bestanden te zoeken die een zekere eigenschap hebben, is er het `find` commando. De meest belangrijke opties hiervan zijn *op naam* (`find -name`) of *op grootte* (`find -size`) te zoeken. Bijvoorbeeld geeft `find . -size +200` alle bestanden die groter dan 200 blokken van 512 byte zijn, dus groter dan 100kB. Hierbij loopt `find` ook over alle subdirectories van het actuele directory.

- Maak een lijst van jou bestanden die een naam met een `x` erin hebben. Hiervoor is het handig dat je ook bij `find` reguliere uitdrukkingen mag gebruiken, bijvoorbeeld de wildcard `*`. Ook hier moeten reguliere uitdrukkingen in *quotes* ingesloten worden.

#### Opdracht 4. Bestanden comprimeren

Om disk ruimte te besparen kan men bestanden die niet actueel benodigd worden, comprimeren. Hiervoor zijn er de commando's `compress` en `gzip`. Met `compress file` wordt het bestand `file` door `file.Z` vervangen, met `gzip file` door `file.gz`. De omgekeerde commando's die het bestand weer uit pakken zijn `uncompress` en `gunzip`. Hierbij maakt het niets uit of het bestand met of zonder suffix aangegeven wordt, dus `gzip file` en `gzip file.gz` werken allebij. In principe is `compress` door `gzip` overbodig gemaakt, want `gzip` kan ook door `compress` gecomprimeerde bestanden uit pakken (maar andersom niet).

- Pas de commando's `compress` en `gzip` op verschillende bestanden toe en ga na hoe sterk de bestanden gecomprimeerde worden. Meestal werkt het veel beter op tekst bestanden dan op data bestanden (bijvoorbeeld plaatjes of `.pdf`-bestanden) en is `gzip` iets beter dan `compress`.
- Sommige programma's zijn zo slim dat ze een gecomprimeerd bestand automatisch (tijdelijk) uit pakken. Dit geldt voor het `gv` programma, waarmee ook gecomprimeerde postscript-bestanden kunnen worden bekeken. Probeer dit uit.

#### Opdracht 5. Spelling

Bij het schrijven van documenten sluipen altijd tik- en spelfouten binnen. Om deze te vinden is er onder UNIX (bijvoorbeeld) het `ispell` commando. Dit is een *interactieve* spelling checker, die bij niet in een intern woordenboek gevonden woorden opties ter verbetering aanbiedt. Er zijn verschillende talen geïnstalleerd, de default is Engels en Nederlands krijgt men met `ispell -d nederlands file`.

- Pas de spelling checker op het bestand `rede.txt` en een eigen bestand toe.

## Checklist

- Ik ken het `man` commando waarmee ik de ingebouwde hulp functie van UNIX oproep en weet hoe ik hiermee informatie kan vinden.
- Ik weet hoe ik de in- en uitvoer van commando's kan omleiden en hoe ik hiermee commando's kan combineren.
- Ik kan met het `grep` commando naar strings zoeken en weet wat reguliere expressies zijn.
- Ik weet hoe ik met het `find` commando bestanden met zekere eigenschappen kan vinden.
- Ik kan disk ruimte besparen door bestanden te comprimeren en weet hoe ik gecomprimeerde bestanden weer uit kan pakken.
- Ik kan met het `ispell` commando de spelling in een bestand checken.