

#KLOOIENMETCOMPUTERS

ARNOUT VAN KEMPEN OVER ROMMELEN IN EEN DIGITALE WERELD

Nu we het bestandssysteem van Linux een beetje hebben bekeken, wordt het tijd voor wat commando's waarmee je niet alleen iets kan zien, maar ook iets kan doen. Opnieuw is de uitnodiging zoveel mogelijk uit te proberen, maar let nu wel op: we gaan dingen uitproberen waarmee het goed mis kan gaan. Daarom eerst een paar commando's om de veiligheid van onze experimenten te vergroten:

Mkdir is het commando om een nieuwe directory te maken. Gebruik dit om een experimenteerhoekje te maken. Alles wat daar verder wordt gemaakt, veranderd of vernietigd, is ongevaarlijk. Zelf maak ik voor dit soort doelen graag een directory 'Zandbak', de ultieme plek om als een spelend kind vies te worden en wat te leren.

Pwd staat voor *present working directory*. Als je het commando geeft, zie je waar je precies bent. Als het antwoord zoiets is als '/home/pi/zandbak' dan weet je dat je nog in de zandbak-directory zit.

Wat gaan we eerst doen? Een eerste bestand maken! Het simpelste commando om een leeg bestand mee te maken is **touch**. Dit commando zet bij bestaande bestanden de datum en tijd naar nu, maar als je het wilt gebruiken op een niet bestaand bestand, wordt het aangemaakt. En als je wilt, maak je

meteen meerdere bestanden. Probeer bijvoorbeeld eens **touch file1 file2 file3** en bekijk dan het resultaat met **ls**.

Linux beschikt over een bijzonder krachtig hulpmiddel, *pipes* en *redirection*. Hiervoor worden de symbolen |, <, > en >> gebruikt. De simpelste twee zijn de laatste. Hiermee wordt de output van een programma of commando naar een bestand gestuurd. Met > een nieuw bestand, en met >> wordt de output toegevoegd. Probeer maar eens **ls > test**. Je ziet niet direct iets gebeuren, maar als je nu met **ls** kijkt, zie je dat het bestand 'test' is ontstaan. De inhoud hiervan kunnen we bekijken met **cat**. Als je nu **cat test** invoert zie je de inhoud van het bestand 'test'. Als het goed is, herken je dat deze inhoud gelijk is aan de inhoud van de directory vóór we **ls > test** als opdracht gaven. Doe je nu **ls >> test** en opnieuw **cat test**, dan zie je dat de nieuwe inhoud na de oude inhoud is toegevoegd.

Met < kunnen we op vergelijkbare manier de input van het toetsenbord vervangen door input uit een bestand. We hebben nog geen commando's bekeken waar dit zin heeft, dus ik laat deze even voor wat het is. Wel wil ik de *pipe* nog laten zien, die met | wordt gemaakt. Hierbij wordt de output van het eerste commando gebruikt als input voor het volgende commando. Om de werking te

laten zien is het nodig dat we wat bestanden in onze directory hebben. Inmiddels zouden in ieder geval file1, file2, file3 en test moeten bestaan.

Voor de *pipes* die we gaan maken gebruiken we twee commando's: **head** en **tail**. Deze programma's leveren de eerste of de laatste regels die worden ingevoerd. Als losse commando's heb je er weinig aan, maar als de input uit een bestand komt, wordt het ineens zinvol. Het gebruik van *pipes* en *redirection* is lastig uit te leggen, maar eenvoudig te begrijpen als je ze aan het werk ziet. Daarom gaan we de volgende commando's invoeren, met de opdracht: bekijk de output en probeer te begrijpen wat je ziet. Als je het begrijpt, probeer dan zelf een *pipe* te bouwen die het *derde* bestand uit de directory toont.

Ls

Ls | head -2

LS | tail -2

LS | head -4 | tail -2

LS | head -4 | tail -2 > pipe

Cat pipe

Succes! ←