Werkcollege 2

8. Piping en tekst lezen

Door middel van piping kan de uitvoer van het ene commando als input van het volgende commando gebruikt worden.

Voorbeeld:

```
ls –l /usr/bin/ | more
```

Het eerste commando **ls -l /usr/bin/** geeft een lijst met de volledige inhoud van de directory /usr/bin/, het tweede commando **more** krijgt deze lijst als input en zet deze op het scherm. Probeer dit ook eens met de pager '**less'** in plaats van de pager 'more'.

> Het commando **more** kan dienen als een filter waarmee tekst op het beeldscherm scherm voor scherm bekeken kan worden. Daarnaast kan je het ook gebruiken om de inhoud van een tekstbestand weer te geven. (vb. `more /etc/passwd` geeft je de inhoud van dit bestand per scherm).
> Het commando **less** is een uitgebreide versie van **more** ("less is more"). Een van de grootste voordelen van less is dat je hiermee niet enkel vooruit kan, maar ook terug. Verder leest less niet het hele bestand in voordat het weergegeven wordt. Zeker bij grote bestanden levert dit een grotere snelheid op.
> Je kan de inhoud ook altijd gewoon op het scherm afprinten zonder een pager. Hiervoor kan je het commando cat gebruiken.
> Het commando cat verwacht zoals less en more als argument een bestandsnaam en dient meer voor korte bestanden af te beelden, aangezien de hele inhoud ineens over je scherm komt.
> De bevelen **less** en **more** zijn commando's, zoals reeds vermeld werd, om de inhoud van bestanden af te beelden, maar via een interactief programma. Je kan namelijk lijn per lijn of scherm per scherm de inhoud bekijken en bij less ook nog zoeken naar patronen in de tekst.

Opdracht 1:

Geef alle lijnen uit het bestand /etc/services weer waarin het woord telnet voorkomt. Doe dit als volgt: open met het commando less het bestand /etc/services. Het commando **grep** kan je gebruiken om een regel uit een bestand te filteren. Maak hiervoor gebruik van piping.

> Het commando **grep** heeft als functie het opzoeken van een bepaalde string (opeenvolging van karakters) op te zoeken in een bestand. Argumenten van grep zijn dus de string of het pattern en het bestand. De uitvoer die grep teruggeeft is elke lijn in het bestand waarin het pattern in voorkomt.

Opdracht 2:

Waar we bij de vorige opdracht met grep horizontaal konden filteren, kunnen we met het commando **cut** verticaal filteren (bijvoorbeeld kolommen eruit filteren). Bekijk het bestand **/etc/passwd**. Voor deze oefening is het de bedoeling dat we ENKEL de gebruikersnamen eruit filteren. Zoek in de manpage op hoe we dit kunnen doen.

TIP: zoek naar de termen **delimiter** en **fields**.

We hebben hierboven met less, more, cat, cut en grep een aantal commando's gezien waarmee je data op je scherm kan lezen. Als laatste zijn er nog 2 extra commando's zeer handig, vooral voor het lezen van de logfiles, **head** en **tail**.
Ben je enkel geïnteresseerd in het x eerste of de y laatste lijnen van het bestand kan je gebruik maken van de bevelen *head* en *tail*. De syntax is zeer eenvoudig en makkelijk te vinden via een -help of in de manpages.

Opdracht 3:

1. Laat de eerste 5 lijnen van het bestand /var/log/syslog afbeelden.
2. Laat de laatste 10 lijnen van het bestand /var/log/syslog afbeelden.

9. Redirection

Door middel van piping stuur je de output van een commando naar een ander commando. Iets wat hier erg op lijkt is redirection. Met redirection stuur je de output niet naar een ander commando, maar wel naar een bestand. Ook kan als input een bestand gebruikt worden. We maken hiervoor gebruik van de tekens **<** en **>**.

> Stuurt de output van een commando door naar een bestand.

Voorbeeld:

ls –l > lstekst

De output staat nu in het bestand "lstekst" en kan u bekijken met vb. **vi, more**, of **cat.** Let op: als er al een bestand lstekst bestaat, wordt dit overschreven!

>> Stuurt de output naar een bestand en voegt deze achteraan in het bestand toe. Het bestand wordt dus niet overschreven. Dit wordt ook een 'append' genoemd.

< Stuurt de inhoud van een bestand door om deze te gebruiken als input voor een commando.

10. Zoeken van bestanden

10.1 Find

Met het commando **find** kan je bestanden zoeken op naam, grootte, aanmaakdatum en eigenaar. Aan dit commando kunnen 3 soorten opties meegegeven worden:

algemene opties

vb. -maxdepth 3 geeft aan dat het commando maximum 3 directories dieper mag gaan zoeken.

de testen

Hiermee bepaal je aan welke eisen een gevonden bestand moet voldoen.
vb. -user … geeft aan dat het bestand van een gegeven persoon moet zijn.

de acties

Hiermee geef je aan wat er met de gevonden bestanden moet gebeuren.

Good-to-know

vb. -print geeft de lijst van gevonden bestanden op je scherm.
-exec zal extra commando's uitvoeren op de gevonden bestanden.

```
find / -name "a*" -user frank -exec cp {} /test \;
```

Om meerdere commando's uit te voeren op de lijst, zet je deze na –exec van elkaar gescheiden met een ';'.
{} wordt vervangen door het gevonden bestand.
Het einde van de lijst wordt aangegeven met '\;'

Opdracht

Gebruik het commando find om alle bestanden op te zoeken die beginnen met de letters xf en waarvan de gebruiker root de eigenaar is. Kopieer deze naar uw homedirectory. Zorg ervoor dat aan het eind van deze actie een lijst wordt gegeven van alle bestanden die gekopieerd zijn. Waarom krijg je foutmeldingen?

10.2 locate

Een alternatief op find is **locate**. Het voordeel van dit commando is dat het werkt met een index, en dus sneller resultaten kan tonen dan find. Maar daarvoor moet deze index wel eerst aangemaakt worden en up-to-date worden gehouden. Hiervoor kan je het commando **updatedb** gebruiken. Hiervoor zijn wel sudo rechten nodig (hierover later meer). In veel distributies wordt dit echter vaak automatisch gescheduled (ingepland).

Opdracht

Vergelijk eens de tijd die nodig is voor onderstaande commando's (uit te voeren van uit de root (/)).

```
locate syslog
find syslog
```

10. Procesbeheer

Het commando ps

Met behulp van het commando **ps** kan je de processen opvragen die momenteel in uitvoering zijn. Wanneer je ps gebruikt zonder opties zie je enkel de processen die jij als gewone gebruiker uitvoert aan deze terminal of console. Bash is één van die processen.
De informatie die je bij elk proces terugvindt, is:
- PID: het unieke procesnummer van het proces
- TTY: het nummer van de virtuele console of terminal van waarop het proces gestart is
- TIME: hoe lang het proces idle is (niets aan het voeren, wacht bv. op een event)
- CMD: het commando gebruikt om het proces op te starten

Gebruik je **ps –l**, dan vind je al wat meer informatie terug waaronder het UID (User Identifier), een uniek nummer dat elke gebruiker toegewezen krijgt, en de prioriteit (PRI) van het proces. Gebruik je het commando **ps u** zie je niet alleen de processen aan de lokale terminal maar die in het algemeen. Alle processen die de gebruiker momenteel uitvoert dus. Als je de manpagina van ps raadpleegt zie je echter een waaier aan opties. Check eens de verschillende opties. Om bijvoorbeeld alle processen op het hele systeem van elke gebruiker af te beelden kan je **ps auxfw** gebruiken.

Opdracht

Welke gebruikers, naast root, hebben nog processen lopen op je systeem?

Het beëindigen van processen

Het commando **kill** is er om processen een vroege dood te bezorgen. Dit bevel is een programma dat een exit system call stuurt naar een bepaald proces. Kill verwacht als argument het PID van het proces dat je wil 'killen'. Uiteraard kan enkel de eigenaar van een proces dat proces killen. Uitzondering daarop is natuurlijk root.

Het bevel **killall** is er om een hele verzameling processen in 1 commando te beëindigen. Dit bevel verwacht als argument de naam van het bevel waarvan je alle voorkomende processen die met dit bevel zijn opgestart wil beëindigen. killall vi beëindigt bijvoorbeeld alle lopende vi-processen.

Opdracht

Laat het programma VI in de achtergrond van het systeem lopen. Doe dit als volgt: **vi &**
Zoek nu het proces in het systeem op met het commando: ps aux Hoe kan je dit zoeken vereenvoudigen via piping en het grep commando? Kill het vi proces!

Het commando top

Het commando top laat de huidige taken van het systeem zien. Je kan top afsluiten door middel van het intypen van de letter q (letter h laat alle opties zien).
Er bestaat ook een iets uitgebreidere versie van top, htop genaamd. Deze moet echter vaak nog apart worden geïnstalleerd. Hierover later meer.

12. Links (shortcuts)

Een link is een verwijzing naar een bestand. Er bestaan 2 soorten links: de hard link en de symbolic link.
Een **hard link** is een verwijzing naar een interne representatie van een bestand (beschrijving van waar de data juist op de schijf te vinden is, de inode). Een bestandsnaam is aan zo een interne representatie gekoppeld, en een hard link is niet meer dan zo een tweede koppeling. In een tabel worden alle koppelingen bijgehouden. Wanneer een koppeling (hard link) verwijderd wordt, zal de data ook pas echt verwijderd worden als er geen andere koppelingen meer bestaan. Eenmaal een hard link gemaakt, lijkt het erop dat het bestand gewoon gekopieerd werd. Dit is echter niet zo. Bij het maken van een kopie ontstaan er twee bestanden. Een wijziging in het bestand via de ene hard link zal ook zichtbaar zijn bij het bekijken van het bestand via een andere hard link. Bij een kopie zal dit uiteraard niet zo zijn.

Een **symbolic link** is een verwijzing naar een ander bestand. Deze bestanden kunnen zelfs op een andere computer staan. Een symbolic link is echter niet meer dan een snelkoppeling (een shortcut): als u het originele bestand weggooit, worden alle links onbruikbaar. Ze werken ook langzamer dan hard links.

Een **symbolic link** mag gemaakt worden naar een bestand of directory, terwijl een **hard link** linkt naar inodes (plaatsen op de HD).

Bij het aanpassen van de rechten van een **hard link**, worden de rechten van het bestand ook aangepast. Veranderen we de naam van de **hard link**, dan blijft hij werken, bij het een **symbolic link** is dit voor beide acties niet zo.

Achterhalen of een bestand een link is, kan met het commando *ls -l*
Symbolic: We zien achteraan waar de link naar verwijst. Vooraan, bij de permissies, zien we de letter 'l' van 'link'.

Hard link: tussen de permissies en de eigenaar van het bestand zien we een getal, dat is het aantal bestaande hardlinks.

Om een link aan te maken wordt het commando "ln" gebruikt:
ln -s target linkname

Zonder de optie -s wordt er een hardlink gemaakt.

Opdracht

Maak in uw persoonlijke directory een directory *links*. Maak een symbolic en een hardlink aan van enkele bestanden. Bij twijfel, raadpleeg de manpage.