Van DOS/Windows naar Linux HOWTO

Door Guido Gonzato, `ggonza at tin.it`,
Vertaald door: Ellen Bokhorst `bokkie at nl.linux.org`

Versie 1.3.5. 31 augustus 2000

Deze HOWTO is opgedragen aan alle (spoedig voormalige?) DOS en Windows gebruikers die besloten hebben naar Linux, de vrije UNIX-kloon, over te stappen. Het doel van dit document is de lezer te helpen zijn of haar kennis van DOS en Windows naar de Linux omgeving om te zetten, als ook in het voorzien van hints in het uitwisselen van bestanden en bronnen tussen de twee OS'sen.

Inhoudsopgave

1 Introductie

1.1 Is Linux iets voor jou?

Laten we politiek correct beginnen. In dit document heb ik het over "Linux", maar ik bedoel "GNU/Linux". Ga alsjeblieft naar `<http://www.gnu.org/gnu/linux-and-gnu.html>` om te zien waarom.

Wil je van de DOS/Windows wereld naar Linux overstappen? Goed idee: Linux is technisch gezien superieur aan DOS, Windows 9x en zelfs Windows NT. Maar pas op: misschien dat het niet van nut is voor je. Dit zijn de belangrijkste verschillen tussen DOS/Windows en Linux:

- Onder Windows draaien Microsoft Office en heel veel spelletjes; wordt waargenomen als eenvoudig te installeren en configureren; het is opmerkelijk onstabiel; heeft een magere performance; crasht frequent.
- Onder Linux draait StarOffice, heel veel technische software en minder spellen; het kan lastig te installeren en configureren zijn; is rotsvast; feilloze performance; crashes komen zelden voor.

Het is aan jou uit te maken wat je nodig hebt. Bovendien geeft Linux je kracht, maar het vraagt wat tijd te leren hoe het te benutten. Dus als je voornamelijk commerciële software nodig hebt, of als je er niets voor voelt om nieuwe commando's en begrippen te leren, kun je je beter ergens anders op concentreren. Wees je ervan bewust dat veel nieuwkomers vanwege aanvangsproblemen opgeven.

Er wordt aan gewerkt Linux eenvoudiger in het gebruik te maken, maar *verwacht er niet bedreven in te zijn zonder dat je heel veel documentatie hebt gelezen en het op z'n minst voor een paar maanden hebt gebruikt.* Linux geeft je geen kant en klare oplossingen. Ondanks deze waarschuwingen, ben ik er 100% van overtuigd dat als je het juiste type gebruiker bent, je zeer tevreden zult zijn over Linux. Tussen twee haakjes, Linux + DOS/Win kunnen prima op dezelfde computer voorkomen.

Benodigd voor deze howto: Ik ga ervan uit dat

- je bekend bent met de basiscommando's en concepten van DOS;
- Linux mogelijk met X Window System (afgekort tot X11) op de juiste wijze op je PC is geïnstalleerd;
- je shell (het equivalent van `COMMAND.COM`) `bash` is.

Tenzij anders aangegeven, wordt bij dit werk uitgegaan van dat 'slechte ouwe' DOS. Her en der wordt informatie over Windows gegeven, maar houd in gedachten dat Windows en Linux totaal verschillend zijn, anders dan DOS dat een of andere sobere relatie met UNIX heeft.

Bedenk je alsjeblieft ook dat dit werkstuk noch een complete inleiding noch een configuratiegids is!

De laatste versie van dit document is in verscheidene formaten beschikbaar op `<http://www.linuxdoc.org>`.

1.2 Vertel verder

Je installeerde Linux en de programma's die je nodig had op je PC. Je gaf jezelf een account (als dit niet zo is, typ dan *nu!* `adduser jenaam` en Linux is opgestart. Je hebt zojuist je naam en wachtwoord ingevuld, en nu kijk je naar het scherm en denkt: "Tja, en wat nu?"

Wanhoop niet. Je bent bijna zover dat je dezelfde dingen doet die je gewend bent met DOS/Win te doen en nog wel meer ook. Als je nu onder DOS/Win zou werken in plaats van onder Linux, dan zou je één van de volgende taken aan het verrichten zijn:

- het draaien van programma's, aanmaken, kopiëren, bekijken, verwijderen, afdrukken hernoemen van bestanden;
- CD'en, MD'en, RD'en, en DIR'ren naar je directory's;
- formatteren van diskettes en er bestanden van/naar kopiëren;
- aanpassen van het systeem;
- surfen op het Internet;
- het schrijven van .BAT bestanden en programma's in je favoriete taal;
- de resterende 1%.

Het zal je verheugen te weten dat deze taken onder Linux op een vergelijkbare manier als onder DOS kunnen worden bewerkstelligd. Onder DOS, gebruikt de gemiddelde gebruiker een paar van de 100+ beschikbare commando's: hetzelfde geldt tot op zekere hoogte voor Linux.

1.2.1 Inleidende begrippen

Oefening baart kunst. Je wordt aangemoedigd om met Linux te experimenteren en spelen: tenzij je als "root" inlogt, kun je het systeem op die manier niet beschadigen. Een paar aandachtspunten:

- ten eerste hoe je Linux veilig af kunt sluiten. Als je een tekstmode scherm ziet, druk dan op <CTRL-ALT-DEL>, wacht totdat het systeem herstart, en zet dan de PC uit. Als je onder het X Window Systeem werkt, druk dan eerst op <CTRL-ALT-BACKSPACE>, daarna <CTRL-ALT-DEL>. Schakel de PC *Nooit* direct uit en maak ook geen gebruik van de reset knop: dit zou het bestandssysteem kunnen beschadigen;
- in tegenstelling tot DOS of Windows, heeft Linux een ingebouwd beveiligingsmechanisme. Met bestanden en directory's zijn permissies geassocieerd; als resultaat daarvan kunnen er een aantal niet door de gewone gebruiker worden benaderd; (zie Sectie 4.3 (Permissies en Eigenaarschap)). DOS en Windows laten je daarentegen de volledige inhoud van je harddisk om zeep helpen;
- er is een speciale gebruiker genaamd "root": de systeembeheerder, met de kracht van leven en dood op je computer. Als je op een eigen PC werkt, zal je ook root zijn. Als root werken is *gevaarlijk*: iedere vergissing kan je systeem net als met DOS/Windows ernstig beschadigen of ruïneren. Werk niet als root tenzij dit absoluut noodzakelijk is;
- veel van de complexiteit van Linux komt door zijn extreme configuratiemogelijkheden: praktisch elke feature en iedere applicatie kan via één of meer configuratiebestanden worden aangepast. Complexiteit is de prijs die moet worden betaald voor kracht;
- omleiding en het door middel van een pipe-symbool doorgeven van gegevens zijn een zijdelingse DOS-feature, een zeer belangrijke en onder Linux veel krachtiger. Eenvoudige commando's kunnen worden samengevoegd om complexe taken te volbrengen. Ik adviseer je deze commando's te leren gebruiken.

1.2.2 Hulp verkrijgen

Er zijn veel manieren om hulp te krijgen met Linux. De belangrijkste zijn:

- *het lezen van documentatie*—Ik meen het. Alhoewel de HOWTO die je aan het lezen bent als een introductie in Linux kan dienen, zijn er verscheidene boeken die je echt zou moeten lezen: op z'n minst Matt Welsh's "Linux Installation and Getting Started" (`<http://www.linuxdoc.org/LDP/gs/gs.html>`) en de Linux FAQ (`<http://www.linuxdoc.org/FAQ/Linux-FAQ/index.html>`). Voel je schuldig net zolang totdat je er op z'n minst één van hebt gelezen;
- de documentatie van de packages die op de computer zijn geïnstalleerd zijn vaak te vinden in subdirectory's onder `/usr/doc/`;
- typ `help` of, beter, `man bash` of `info bash` voor het verkrijgen van hulp over de "interne commando's" van de shell;
- type voor hulp met een commando `man commando` waarmee de handleiding ("man") page van `commando` wordt aangeroepen. Typ als alternatief `info commando` waarmee de info page die betrekking heeft op `commando` wordt aangeroepen; info is een op hypertekst gebaseerd documentatiesysteem, misschien niet intiütief om als eerste te gebruiken. Tenslotte kun je als laatste `apropos commando` of `whatis commando` gebruiken. Voor al deze commando's geldt dat je op 'q' drukt om het helpsysteem te verlaten.

- als laatste op het Internet: de juiste plaats voor het verkrijgen van hulp is Usenet, zoals `<news:comp.os.linux.setup>`. Email mij alsjeblieft niet om hulp want ik ben zeer overbelast.

1.3 Afspraken

In dit hele document zullen voorbeelden vaak het volgende formaat volgen: `<...>` is een vereist argument, en `[...]` is een optioneel argument. Voorbeeld:

```
$ tar -tf <bestand.tar> [> omleidingsbestand]
```

`bestand.tar` moet worden opgegeven, maar de omleiding naar het `omleidingsbestand` is optioneel.

"LMP" betekent "Lees alsjeblieft de Man Pages voor verdere informatie". Ik kan niet genoeg benadrukken hoe belangrijk het lezen van documentatie is.

Als de prompt van een voorbeeldcommando een `#` is, kan het commando alleen door root worden uitgevoerd.

2 Voor de ongeduldigen

Wil je graag beginnen? Bekijk dan de volgende tabel:

```
DOS                     Linux                   Opmerkingen
------------------------------------------------------------------------------

ATTRIB (+-)attr bestand chmod <mode> bestand    totaal anders
BACKUP                  tar -Mcvf device dir/   idem
CD dirnaam\             cd dirnaam/             bijna dezelfde syntax
COPY bestand1 bestand2  cp bestand1 bestand2    idem
DEL bestand             rm bestand              let op - geen undelete
DELTREE dirnaam         rm -R dirnaam/          idem
DIR                     ls                      niet exact dezelfde syntax
DIR bestand /S          find . -naam bestand    totaal anders
EDIT bestand            vi bestand              Ik denk dat je 't niet leuk vindt
                        jstar bestand           voelt als dos' edit
EDLIN bestand           ed bestand              vergeet het
FORMAT                  fdformat,
                        mount, umount           heel andere syntax
HELP commando           man commando,           zelfde filosofie
                        info commando
MD dirnaam              mkdir dirnaam/          bijna dezelfde syntax
MORE < bestand          less bestand            veel beter
MOVE bestand1 bestand2  mv bestand1 bestand2    idem
NUL                     /dev/null               idem
PRINT bestand           lpr bestand             idem
PRN                     /dev/lp0,
                        /dev/lp1                idem
RD dirnaam              rmdir dirnaam/          bijna dezelfde syntax
REN bestand1 bestand2   mv bestand1 bestand2    niet voor meerdere bestanden
RESTORE                 tar -Mxpvf device       andere syntax
TYPE bestand            less bestand            veel beter
WIN                     startx                  hemelsbreed verschil!
```

Raadpleeg alsjeblieft de volgende secties als je meer nodig hebt dan een tabel met commando's.

3 Maak kennis met bash

Goed nieuws: onder Linux typ je veel minder achter de prompt, omdat de `bash` shell zo mogelijk voor je typt, en het heeft geweldige regelediting mogelijkheden. Om mee te beginnen, haalt de cursorpijl naar boven de voorgaande commandoregels terug; maar er is meer. Het indrukken van de <TAB> vult bestands- en directorynamen aan, dus het typen van

```
$ ls /uTABloTABbTAB
```

is als het intikken van

```
$ ls /usr/local/bin
```

Als er dubbelzinnigheden optreden, zoals bij het intikken van

```
$ ls /uTABloTABiTAB
```

stopt `bash` omdat het niet weet of je `/usr/local/info` of `/usr/local/include` bedoelt. Voeg meer tekens toe en druk dan nogmaals op <TAB>.

Andere handige toetsaanslagen zijn <ESC-BACKSPACE> waarmee een woord links wordt verwijderd, en <ESC-D> waarmee een woord rechts wordt verwijderd; <ESC-F> verplaatst de cursor één woord naar rechts, <ESC-B> naar links; <CTRL-A> verplaatst de cursor naar het begin van de regel, <CTRL-E> naar het einde. De <ALT> toets is equivalent aan <ESC>.

Genoeg voor het moment. Zodra je aan deze sneltoetsen gewend raakt, zal je de DOS-prompt zeer ergerlijk vinden...

4 Bestanden en programma's

4.1 Bestanden: Opmerkingen vooraf

Linux heeft een directory- en bestandsstructuur die erg veel lijkt op die van DOS/Win. Bestanden hebben namen die aan bepaalde regels moeten voldoen, worden opgeslagen in directory's, een aantal ervan zijn uitvoerbaar, en hiervan hebben de meeste commando-switches. Bovendien kun je gebruik maken van wildcard characters (jokertekens), redirection en piping (het omleiden/doorsturen van uitvoer). Er zijn slechts een paar kleine verschillen:

- onder DOS zijn de bestandsnamen in de zogenoemde 8.3 vorm; b.v. `NOTENOUG.TXT`. Onder Linux kunnen we dat beter. Als je Linux installeerde door gebruik te maken van een bestandssysteem als ext2 of umsdos, kun je langere bestandsnamen gebruiken (tot aan 255 tekens), en met meer dan één punt: bijvoorbeeld: `Dit_is.een.ZEER_lange.bestandsnaam`. Merk alsjeblieft op dat ik zowel gebruik maakte van hoofdletters als kleine letters: in feite...
- worden hoofdletters en kleine letters in bestandsnamen als verschillende tekens gezien. Daarom zijn `FILENAME.tar.gz` en `filename.tar.gz` twee verschillende bestanden. `ls` is een commando, `LS` is een fout;
- Windows gebruikers, let op wanneer je gebruik maakt van bestandsnamen onder Linux. Als er in een bestandsnaam spaties voorkomen (niet aan te bevelen maar wel mogelijk), moet je de bestandsnaam tussen dubbele aanhalingstekens plaatsen wanneer je ernaar refereert. Bijvoorbeeld:

```
$ # het volgende commando maakt een directory genaamd "Mijn oude bestanden"
$ mkdir "Mijn oude bestanden"
$ ls
Mijn oude bestanden    bin     tmp
```

Bovendien is het beter een aantal tekens niet te gebruiken: een aantal daarvan zijn `!*$&#`.

- er zijn voor programma's geen verplichte extensies zoals .COM en .EXE of .BAT voor batchbestanden. Uitvoerbare bestanden worden aan het einde van de bestandsnaam door een asterisk '`*`' gemarkeerd wanneer je het commando `ls -F` geeft. Bijvoorbeeld:

```
$ ls -F
Ik_ben_een_dir/   cindy.jpg    cjpg*   brief_aan_Joe    mijn_1e_script*  oud~
```

De bestanden `cjpg*` en `mijn_1e_script*` zijn uitvoerbare bestanden, dat wil zeggen "programma's". Onder DOS eindigen backupbestanden op .BAK, onder Linux eindigen ze op een tilde '`~`'. Een bestand waarvan de naam begint met een punt wordt als verborgen aangemerkt. Voorbeeld: het bestand `.Ik.ben.een.verborgen.bestand` zal na het commando `ls` niet worden getoond;

- DOS programma switches worden verkregen met `/switch`, Linux switches met `-switch` of `-switch`. Voorbeeld: `dir /s`<tt> wordt `ls -R`. Merk op dat veel DOS programma's, zoals `PKZIP` of `ARJ`, switches in de stijl van UNIX gebruiken.

Je kunt nu naar de sectie 4.4 (Commando's van DOS naar Linux vertalen) gaan, maar als ik jou was zou ik verder lezen.

4.2 Symbolische links

UNIX heeft een type bestand dat onder DOS niet voorkomt: de symbolische link. Je kunt hierbij denken aan een verwijzer naar een bestand of directory, en het kan in plaats van het bestand of directory waarnaar het verwijst worden gebruikt; het is vergelijkbaar met de snelkoppelingen van Windows. Voorbeelden van symbolische links zijn `/usr/X11`, die naar `/usr/X11R6` verwijst; `/dev/modem`, die naar `/dev/ttyS0` of `/dev/ttyS1` verwijst.

Om een symbolische link aan te maken:

```
$ ln -s <bestand_of_dir> <linknaam>
```

Voorbeeld:

```
$ ln -s /usr/doc/g77/DOC g77manual.txt
```

Nu kun je naar `g77manual.txt` verwijzen in plaats van `/usr/doc/g77/DOC`. Links verschijnen als volgt in directorylistings:

```
$ ls -F
g77manual.txt@
$ ls -l
(van alles en nog wat...)          g77manual.txt -> /usr/doc/g77/DOC
```

4.3 Permissies en Eigenaarschap

DOS bestanden en directory's hebben de volgende kenmerken: A (archive = archief), H (hidden = verborgen), R (read-only = alleen lezen), en S (system = systeem). Alleen H en R hebben onder Linux een betekenis: verborgen bestanden beginnen met een punt, en wat betreft het kenmerk R, lees verder.

Onder UNIX heeft een bestand "permissies" en een eigenaar, die op zijn beurt weer aan een "groep" toebehoort. Kijk naar dit voorbeeld:

```
$ ls -l /bin/ls
-rwxr-xr-x  1  root  bin  27281 Aug 15 1995 /bin/ls*
```

In het eerste veld staan de permissies van het bestand `/bin/ls`, welke aan root toebehoort, groep bin. De resterende informatie latend voor wat het is, onthoud dat `-rwxr-xr-x` van links naar rechts betekent:

- het bestandstype is (`-` = gewoon bestand, `d` = directory, `l` = link, enz.); `rwx` zijn de permissies voor de eigenaar van het bestand (read/lezen, write/schrijven, execute/uitvoeren); `r-x` zijn de permissies voor de groep van de eigenaar van het bestand (read, execute); (Ik ga het niet hebben over het begrip groep, je kunt zolang je beginner bent zonder ;-0 `r-x` zijn de permissies voor alle andere gebruikers (read, execute).

De directory `/bin` heeft ook permissies: zie sectie 5.2 (Directory Permissies) voor verdere details. Daarom kun je het bestand `/bin/ls` niet verwijderen tenzij je root bent: je hebt geen permissie om dat te doen. Het commando voor het wijzigen van de bestandspermissies is:

```
$ chmod <wieXperm> <file>
```

'wie' is `u` (user, dat is de eigenaar), `g` (group), `o` (other=anderen), X is of `+` of `-`, perm is `r` (read), `w` (write), of `x` (execute). Algemene voorbeelden van `chmod` zijn:

```
$ chmod +x bestand
```

hiermee wordt de execute permissie voor het bestand ingesteld.

```
$ chmod go-rw bestand
```

hiermee wordt de read en write permissies voor iedereen behalve de eigenaar ingesteld.

```
$ chmod ugo+rwx bestand
```

dit geeft iedereen read, write en execute permissie.

```
# chmod +s bestand
```

hiermee wordt een zogenoemd "setuid" of "suid" bestand aangemaakt— een bestand dat iedereen uit mag voeren met de privileges van de eigenaar. Je zal ongetwijfeld root suid bestanden tegenkomen; dit zijn vaak belangrijke systeembestanden, zoals de X-server.

Een beknoptere manier om naar permissies te refereren is met cijfers: `rwxr-xr-x` kan worden uitgedrukt als 755 (iedere letter correspondeert met een bit: `--` is 0, `-x` is 1, `-w-` is 2, `-wx` is 3...). Het ziet er moeilijk uit, maar met wat oefening zal je het concept begrijpen. root, als superuser, kan de bestandspermissies van iedereen wijzigen. LMP.

4.4 Bestanden: Commando's vertalen

Links de DOS-commando's; rechts de Linux tegenhanger.

```
ATTRIB:         chmod
COPY:           cp
DEL:            rm
MOVE:           mv
REN:            mv
TYPE:           more, less, cat
```

Omleidings- en andere operatoren: < > >> |

Wildcards: `* ?`

`nul:` `/dev/null`

`prn, lpt1:` `/dev/lp0 of /dev/lp1; lpr`

4.4.1 Voorbeelden

```
DOS                                     Linux
---------------------------------------------------------------------

C:\GUIDO>ATTRIB +R FILE.TXT             $ chmod 400 file.txt
C:\GUIDO>COPY JOE.TXT JOE.DOC           $ cp joe.txt joe.doc
C:\GUIDO>COPY *.* TOTAL                 $ cat * > total
C:\GUIDO>COPY FRACTALS.DOC PRN          $ lpr fractals.doc
C:\GUIDO>DEL TEMP                       $ rm temp
C:\GUIDO>DEL *.BAK                      $ rm *~
C:\GUIDO>MOVE PAPER.TXT TMP\            $ mv paper.txt tmp/
C:\GUIDO>REN PAPER.TXT PAPER.ASC        $ mv paper.txt paper.asc
C:\GUIDO>PRINT LETTER.TXT               $ lpr letter.txt
C:\GUIDO>TYPE LETTER.TXT                $ more letter.txt
C:\GUIDO>TYPE LETTER.TXT                $ less letter.txt
C:\GUIDO>TYPE LETTER.TXT > NUL          $ cat letter.txt > /dev/null
        n/a                             $ more *.txt *.asc
        n/a                             $ cat section*.txt | less
```

Opmerkingen:

- `*` is slimmer onder Linux: `*` komt overeen met alle bestanden behalve de verborgen bestanden; `.*` komt overeen met alle verborgen bestanden (maar ook de huidige directory '`.`' en parent directory '`..`': let op!); `*.*` komt alleen overeen met bestanden met een '`.`' in het midden of die op een punt eindigen; `p*r` komt overeen met zowel 'peter' als 'piper'; `*c*` komt overeen met zowel 'picked' als 'peck';
- druk bij gebruik van `more`, op de <SPATIEBALK> om door het bestand te bladeren, 'q' om het te verlaten. `less` is intuïtiever en laat je de pijltjestoetsen gebruiken;
- er is geen `UNDELETE`, dus *bedenk je tweemaal* voordat je iets verwijdert;
- in aanvulling op de < > >> van DOS, heeft Linux `2>` om foutmeldingen om te leiden (stderr); bovendien wordt met `2>&1` stderr naar stdout omgeleid, en `1>&2` leidt stdout om naar stderr;
- Linux heeft nog een wildcard: de `[]`. Gebruik: `[abc]*` komt overeen met bestanden die beginnen met een a, b, c; `*[I-N1-3]` komt overeen met bestanden die eindigen op I, J, K, L, M, N, 1, 2, 3;

- `lpr` <bestand> drukt een bestand af in de achtergrond. Gebruik `lpq` voor het controleren van de status van de afdrukwachtrij; gebruik `lprm` voor het verwijderen van een bestand uit de afdrukwachtrij;
- er is geen op DOS lijkend `RENAME`; dat wil zeggen dat `mv *.xxx *.yyy` niet zal werken. Een op REN lijkend commando is te vinden op `<ftp://metalab.unc.edu/pub/Linux/utils/file>`;
- gebruik `cp -i` en `mv -i` om te worden gewaarschuwd wanneer een bestand zal worden overschreven.

4.5 Programma's uitvoeren: Multitasking en Sessies

Om een programma uit te voeren, typ je net als onder DOS de naam ervan in. Als de directory (Sectie 5 (Gebruik van directory's)) waarin het programma is opgeslagen in het PATH is opgenomen (Sectie 8.1 (Systeem initialisatiebestanden)), zal het programma worden opgestart. Uitzondering: in tegenstelling tot DOS, zal een programma in de huidige directory niet worden uitgevoerd, tenzij de directory in het PATH is opgenomen. Je kunt hieraan ontsnappen: als `prog` je programma is, typ dan `./prog`.

Zo ziet een typische commandoregel eruit:

```
$ commando [-s1 [-s2] ... [-sn]] [par1 [par2] ... [parn]] [< invoer] [> uitvoer]
```

waar de `-s1`, ..., `-sn` staan voor programma-switches. `par1`, ..., `parn` zijn de programma parameters. Je kunt verscheidene commando's op de commandoregel aanroepen:

```
$ commando-1 ; commando-2 ; ... ; commando-n
```

Dat is alles over het draaien van programma's, maar het is eenvoudig een stap verder te gaan. Eén van de belangrijkste redenen om Linux te gebruiken is dat het een multitasking os is— het kan verscheidene programma's tegelijkertijd draaien (van nu af aan, processen). Je kunt processen in de achtergrond opstarten en gewoon door blijven werken. Bovendien staat Linux toe dat je meerdere sessies hebt lopen: het is alsof je veel computers tegelijk aan het werk zet!

- Druk op <ALT-F1> ... <ALT-F6> om naar sessie 1..6 onder de virtuele console over te schakelen
- Typ `su - <loginnaam>` voor het starten van een nieuwe sessie onder dezelfde v.c. zonder de huidige v.c. te verlaten. Voorbeeld: `su - root`. Dit komt bijvoorbeeld van pas als je een taak uit moet voeren waarvoor alleen de root-gebruiker permissie heeft.
- Typ voor het beëindigen van een sessie `exit`. Als er gestopte jobs zijn (zie verderop), krijg je een waarschuwing.
- Voeg een ampersand '`&`' aan het einde van de commandoregel toe voor het in de achtergrond opstarten van een proces:

  ```
  $ prognaam [-switches] [parameters] [< invoer] [> uitvoer] &
  [1] 123
  ```

  de shell identificeert het proces met een jobnummer (b.v. `[1]`; zie hieronder), en met een PID (Proces Identificatie Nummer; 123 in ons voorbeeld).
- Typ `ps ax` om te bekijken hoeveel processen er zijn. Hiermee krijg je als uitvoer een lijst met de thans draaiende processen.

- Typ `kill` <PID> voor het killen (beëindigen) van een proces. Wellicht dat je het eens nodig hebt als je niet weet hoe je een proces op de juiste wijze moet laten eindigen... Tenzij je root bent, kun je andermans processen niet killen. Soms kan een proces alleen worden beëindigd met `kill -SIGKILL <PID>`.

  Bovendien laat de shell je een proces stoppen of tijdelijk uitstellen, een proces naar de achtergrond sturen, en een proces vanuit de achtergrond naar de voorgrond brengen. In deze context, worden processen "jobs" genoemd.

- Typ `jobs` om te zien hoeveel jobs er zijn. Hier worden de jobs geïndentificeerd door een jobnummer, niet via een PID.
- Druk op <CTRL-C> om een in de voorgrond draaiend proces te stoppen (dit werkt niet altijd).
- Druk op <CTRL-Z> om een in de voorgrond draaiend proces uit te stellen (idem).
- Typ `bg` <%job> voor een uitgesteld proces naar de achtergrond te sturen (het wordt een job).
- Voor het naar de voorgrond brengen van een job typ je `fg` <%job>. Voor het naar de voorgrond brengen van de laatste job die naar de achtergrond is gestuurd, typ je eenvoudigweg `fg`.
- Typ voor het killen van een job `kill` <%job> waarbij <job> gelijk kan zijn aan 1, 2, 3,...

Met deze commando's kun je tegelijkertijd een disk formatteren, een boel bestanden zippen, een programma compileren en een archief uitpakken en nog steeds de prompt tot je beschikking hebben. Probeer dit eens met Windows, enkel om het verschil in performance te zien (als het niet crasht, uiteraard).

4.6 Programma's op Remote computers uitvoeren

Voor het uitvoeren van een programma op een remote computer met de naam `remote.machine.edu`:

```
$ telnet remote.machine.edu
```

Start na het inloggen je favoriete programma. Onnodig te zeggen dat je een shell-account op de remote computer nodig hebt.

Als je X11 hebt, kun je zelfs een X-applicatie op een remote computer uitvoeren en het op je X scherm weergeven. Stel dat `remote.machine.edu` de remote X-computer is en `local.linux.box` je Linux-computer. Om een X-programma vanaf `local.linux.box` dat zich bevindt op `remote.machine.edu` uit te voeren, doe je het volgende:

- start X11 op, start een `xterm` of equivalente terminal emulator, en typ dan:

  ```
  $ xhost +remote.machine.edu
  $ telnet remote.machine.edu
  ```

- typ na het inloggen:

  ```
  remote:$ DISPLAY=local.linux.box:0.0
  remote:$ prognaam &
  ```

  (in plaats van `DISPLAY...`, moet je mogelijk opgeven: `setenv DISPLAY local.linux.box:0.0`. Dit is afhankelijk van de remote shell.)

En ziedaar! Nu zal `prognaam` op `remote.machine.edu` worden opgestart en op je computer worden weergegeven. Probeer dit echter niet via een modem, aangezien dit zo langzaam is dat het niet meer werkbaar is. Bovendien is dit een grove en onveilige methode: lees alsjeblieft de "Remote X Apps mini-HOWTO" op `<http://www.linuxdoc.org/HOWTO/mini/Remote-X-Apps.html>`.

5 Gebruik van directory's

5.1 Directory's: Opmerkingen vooraf

We hebben tussen bestanden onder DOS/Win en Linux verschillen gezien. Wat betreft directory's is de root directory onder DOS/Win \, onder Linux is het /. Evenzo worden geneste directory's onder DOS/Win gescheiden door een \ onder Linux door een /. Voorbeelden van directorypaden:

```
DOS:    C:\PAPERS\GEOLOGY\MID_EOC.TEX
Linux:  /home/guido/papers/geology/middle_eocene.tex
```

Zoals gebruikelijk is `..` de parent directory en `.` de huidige directory. Onthoud dat het systeem je niet toelaat overal waar je maar wil `cd`, `rd`, of `md` uit te voeren. Iedere gebruiker heeft zijn of haar zaakjes in een directory genaamd 'home', gegeven door de systeembeheerder; op mijn PC bijvoorbeeld, is mijn homedir `/home/guido`.

5.2 Directory permissies

Ook directory's hebben permissies. Wat we in paragraaf 4.3 (Permissies en Eigenaarschap) hebben gezien, geldt ook voor directory's (user, group, en other). Voor een directory betekent `rx` dat je met `cd` naar die directory kunt gaan, en `w` betekent dat je een bestand in de directory of de directory zelf kunt verwijderen (overeenkomstig de permissies van het bestand uiteraard).

Om bijvoorbeeld te voorkomen dat andere gebruikers in `/home/guido/text` rond gaan snuffelen:

```
$ chmod o-rwx /home/guido/text
```

5.3 Directory's: Commando's vertalen

```
DIR:            ls, find, du
CD:             cd, pwd
MD:             mkdir
RD:             rmdir
DELTREE:        rm -rf
MOVE:           mv
```

5.3.1 Voorbeelden

```
DOS                                     Linux
---------------------------------------------------------------------

C:\GUIDO>DIR                            $ ls
C:\GUIDO>DIR FILE.TXT                   $ ls file.txt
C:\GUIDO>DIR *.H *.C                    $ ls *.h *.c
C:\GUIDO>DIR/P                          $ ls | more
C:\GUIDO>DIR/A                          $ ls -l
C:\GUIDO>DIR *.TMP /S                   $ find / -name "*.tmp"
C:\GUIDO>CD                             $ pwd
        n/a - zie opmerking             $ cd
        idem                            $ cd ~
        idem                            $ cd ~/temp
C:\GUIDO>CD \OTHER                      $ cd /other
```

```
C:\GUIDO>CD ..\TEMP\TRASH           $ cd ../temp/trash
C:\GUIDO>MD NEWPROGS                $ mkdir newprogs
C:\GUIDO>MOVE PROG ..               $ mv prog ..
C:\GUIDO>MD \PROGS\TURBO            $ mkdir /progs/turbo
C:\GUIDO>DELTREE TEMP\TRASH         $ rm -rf temp/trash
C:\GUIDO>RD NEWPROGS                $ rmdir newprogs
C:\GUIDO>RD \PROGS\TURBO            $ rmdir /progs/turbo
```

Opmerkingen:

- bij gebruik van `rmdir`, moet de te verplaatsen directory leeg zijn. Gebruik `rm -rf` voor het verwijderen van een directory met gehele inhoud (op eigen risico).
- het teken '`~`' is een verkorte vorm voor de naam van je homedirectory. De commando's `cd` en `cd ~` zullen je naar je homedirectory laten terugkeren van waar je je ook bevindt; het commando `cd ~/tmp` brengt je naar `/home/je_home/tmp`.
- `cd -` "annuleert" de laatste `cd`.

6 Floppy's, Harddisks, en dergelijke

Er zijn onder Linux twee manieren om devices te beheren: op de manier van DOS en op de manier van Unix. Maak je keuze.

6.1 Devices beheren zoals DOS het doet

In de meeste Linux distributies is de Mtools suite opgenomen, een set commando's welke perfect equivalent is aan de DOS tegenhanger, maar waarvan de commando's met een 'm' beginnen: d.w.z., `mformat`, `mdir`, `mdel`, `mmd`, enzovoort. Ze kunnen zelfs lange bestandsnamen behouden, maar niet de bestandspermissies. Als je Mtools configureert door het wijzigen van een bestand genaamd `/etc/mtools.conf` (een voorbeeld is in de distributie opgenomen), kun je ook de DOS/Win partitie, de CD-ROM en de Zip-drive benaderen. Voor het formatteren van een nieuwe disk is het `mformat` commando echter niet geschikt. Als root zal je het volgende commando van te voren uit moeten voeren: `fdformat /dev/fd0H1440`.

Je kunt bestanden op de diskette niet benaderen met een commando zoals bv `less a:file.txt`! Dit is het nadeel van het benaderen van disks op de wijze van DOS.

6.2 Devices beheren zoals Unix het doet

UNIX heeft een andere manier om met devices om te gaan. Er zijn geen aparte volumes zoals A: of C:; een disk, of het nu een diskette is of wat dan ook, wordt onderdeel van het lokale bestandssysteem via een bewerking genaamd "mounting". Wanneer je met de diskette klaar bent, moet je het "unmounten" voordat je de diskette uit het station haalt.

Het fysiek formatteren van een disk is een ding, er een bestandssysteem op maken is weer iets anders. Het DOS commando `FORMAT A:` doet beiden, maar onder Linux zijn er twee aparte commando's voor. Zie hiervoor hoe je een diskette formatteert, het aanmaken van een bestandssysteem gaat zo:

```
# mkfs -t ext2 -c /dev/fd0H1440
```

In plaats van `ext2` kun je `dos`, `vfat` (aanbevolen) of andere formaten gebruiken. Zodra de disk is voorbereid, mount je het met het commando:

```
# mount -t ext2 /dev/fd0 /mnt
```

waarbij je het juiste bestandssysteem opgeeft als je geen gebruik maakt van `ext2`. Nu kun je de bestanden op de diskette adresseren door gebruik te maken van `/mnt` in plaats van A: of B:. Voorbeelden:

```
DOS                                   Linux
---------------------------------------------------------------------

C:\GUIDO>DIR A:                       $ ls /mnt
C:\GUIDO>COPY A:*.*                   $ cp /mnt/* .
C:\GUIDO>COPY *.ZIP A:                $ cp *.zip /mnt
C:\GUIDO>EDIT A:FILE.TXT              $ jstar /mnt/file.txt
C:\GUIDO>A:                           $ cd /mnt
A:> _                                 /mnt/$ _
```

Als je klaar bent, *moet* je voordat je de disk eruit haalt het unmounten met het commando

```
# umount /mnt
```

Uiteraard hoef je de commando's `fdformat` en `mkfs` alleen te gebruiken voor ongeformatteerde disks, niet op voorheen gebruikte disks. Als je station B: wilt gebruiken, refereer je in de voorbeelden hiervoor ernaar met `fd1H1440` en `fd1` in plaats van `fd0H1440` en `fd0`.

Onnodig te zeggen dat wat voor diskettes geldt ook voor andere devices geldt; bijvoorbeeld, wellicht dat je een andere harddisk of CD-ROM station wilt mounten. Zo mount je een CD-ROM:

```
# mount -t iso9660 /dev/cdrom /mnt
```

Dit was de "officiële manier om je disks te mounten, maar er is een andere manier. Aangezien het wat lastig is root te moeten zijn voor het mounten van een diskette of CD-ROM, kan gebruikers worden toegestaan ze op deze wijze te mounten:

- doe als root het volgende:

  ```
  # mkdir /mnt/floppy ; mkdir /mnt/cdrom
  # chmod 777 /mnt/floppy /mnt/cd*
  # # zorg ervoor dat het 't juiste CD-ROM device is
  # chmod 666 /dev/hdb ; chmod 666 /dev/fd*
  ```

- voeg aan `/etc/fstab` de volgende regels toe:

  ```
  /dev/cdrom      /mnt/cdrom  iso9660 ro,user,noauto          0       0
  /dev/fd0        /mnt/floppy vfat    user,noauto             0       0
  ```

Mount nu een DOS diskete en een CD-ROM:

```
$ mount /mnt/floppy
$ mount /mnt/cdrom
```

`/mnt/floppy` en `/mnt/cdrom` kunnen nu door iedere gebruiker worden benaderd. Als dit je wat uitmaakt, denk er dan aan dat het iedereen toestaan op deze manier disks te mounten een gapend beveiligingslek is.

Twee handige commando's zijn `df`, welke informatie geeft over gemounte bestandssystemen, en `du dirnaam` welke de door de directory in beslag genomen diskruimte rapporteert.

6.3 Backup maken

Er zijn verscheidene packages om je te helpen, maar het minste wat je kunt doen voor een multi-volume backup (als root) is:

```
# tar -M -cvf /dev/fd0H1440 dir_to_backup/
```

Zorg ervoor dat er een geformatteerde diskette in het station is en dat je er nog verscheidene bij de hand hebt. Voor het terugzetten doe je de eerste diskette in het diskettestation en geef je op:

```
# tar -M -xpvf /dev/fd0H1440
```

7 En hoe zit het met Windows?

Het "equivalent" van Windows is het grafische systeem X Window System. In tegenstelling tot Windows of de Mac, werd X11 niet voor eenvoudig gebruiksgemak of voor een fraaie weergave ontworpen, maar slechts om te voorzien in grafische mogelijkheden voor UNIX werkstations. Dit zijn de belangrijkste verschillen:

- terwijl Windows er in de gehele wereld hetzelfde uitziet, geldt dit niet voor X11: het is veel beter te configureren. X11's algehele look wordt weergegeven door een keycomponent genaamd "window manager", waarin je een brede keuze hebt: `fvwm`, een basis maar mooi en geheugenefficiënt, `fvwm2-95`, `Afterstep`, `WindowMaker`, `Enlightenment`, en nog heel wat meer. De w.m. wordt gewoonlijk aangeroepen vanuit `.xinitrc`;
- je w.m. kan zo worden geconfigueerd zoals een venster onder Windows fungeert: je klikt erop en het verschijnt op de voorgrond. Een andere mogelijkheid is dat het naar de voorgrond wordt verplaatst als de muis erover beweegt ("focus"). Ook kan het plaatsen van vensters op het scherm automatisch of interactief zijn: als een vreemd frame in plaats van je programma verschijnt, klik dan met de linkermuis op de plaats waar je wilt dat het tevoorschijn komt;
- de meeste features kunnen worden aangepast door het wijzigen van één of meer configuratiebestanden. Lees de docs van je w.m.: het configuratiebestand kan b.v. zijn `.fvwmrc`, `.fvwm2rc95`, `.steprc`, enz. Een voorbeeld configuratiebestand is meestal wel te vinden in `/etc/X11/window-manager-name/system.window-manager-name`;
- X11 applicaties zijn geschreven met behulp van speciale library's ("widget sets"); aangezien er verscheidene beschikbaar zijn, zien de applicaties er anders uit. De meest fundamentele applicaties zijn degenen die gebruik maken van de Athena widgets (2–D look; `xdvi`, `xman`, `xcalc`); anderen maken gebruik van Motif (`netscape`), anderen gebruiken nog steeds Tcl/Tk, Qt, Gtk, XForms, of wat je ook maar hebt. Bijna al deze library's voorzien in ruwweg dezelfde look en feel als Windows;
- de feel kan helaas verwarrend zijn. Als je bijvoorbeeld met behulp van de muis een regel tekst selecteert en de <BACKSPACE> indrukt, verwacht je dat de regel zal verdwijnen, nietwaar? Met op Athena gebaseerde applicaties werkt dit niet, maar dit werkt wel zo met andere widget sets;
- hoe schuifbalken en het wijzigen van de grootte werkt is afhankelijk van de w.m. en de widget set. Tip: als je bemerkt dat de schuifbalken niet zo werken als je verwacht, probeer dan de middelste knop of de twee knoppen samen om ze te verplaatsen;
- applicaties hebben standaard geen ikoon, maar ze kunnen er meerdere hebben. De meeste w.m. hebben een menu wat je kunt openen door op de desktop te klikken ("root window"); onnodig te zeggen dat het menu kan worden aangepast. Voor het wijzigen van de opmaak van het root window gebruik je `xsetroot` of `xloadimage`;

- het klembord kan slechts tekst bevatten, en gedraagt zich vreemd. Zodra je tekst hebt geselecteerd, is het al naar het klembord gekopieerd: verplaats je naar elders en druk voor het plakken op de middelste knop. Er is een applicatie genaamd `xclipboard`, welke in meerdere klembordbuffers voorziet;
- slepen en neerzetten is een optie, en is alleen beschikbaar als je X11 applicaties en/of een w.m. gebruikt dat dit ondersteunt.

Met dit te hebben gezegd is er goed nieuws voor je. Er zijn projecten die zich richten op een zelfde look en feel en het fungeren van X11 als Windows. Gnome, `<http://www.gnome.org>`, en KDE, `<http://www.kde.org>`, boezemen eenvoudigweg ontzag in. Zeer waarschijnlijk maakt je distributie van één of beiden gebruik. Je zal geen spijt meer hebben van je Windows desktop!

8 Aanpassen van het systeem

8.1 Systeem Initialisatie Bestanden

Twee belangrijke bestanden onder DOS zijn `AUTOEXEC.BAT` en `CONFIG.SYS`, die tijdens de systeemstart worden gebruikt om het systeem te initialiseren, een aantal omgevingsvariabelen, zoals PATH en FILES in te stellen, en mogelijk een programma of batchbestand op te starten. Daar bovenop heeft Windows het beruchte register—één van de slechtste ideeën ooit in de computerwetenschap.

Onder Linux zijn er heel veel initialisatiebestanden te vinden, waar je van een aantal maar beter af kunt blijven totdat je exact weet wat je aan het doen bent; ze zijn te vinden in de `/etc` directory. Alle configuratie kan worden gedaan door het wijzigen van gewone tekstbestanden. Als je alleen het PATH en andere omgevingsvariabelen hoeft in te stellen, of je de loginberichten wilt wijzigen of bij het inloggen automatisch een programma op wilt laten starten, kijk dan eens naar de volgende bestanden:

```
BESTANDEN                            OPMERKINGEN

/etc/issue                           instellen bericht voor inloggen
/etc/motd                            instellen bericht na inloggen
/etc/profile                         instellen $PATH en andere variabelen,enz
/etc/bashrc                          instellen aliassen en functies, enz
/home/je_home/.bashrc                instellen van je aliassen + functies
/home/je_home/.bash_profile   of
/home/je_home/.profile               instellen omgeving + starten van
                                     programma's
```

Als het laatste bestand voorkomt (het is een verborgen bestand), zal het na het inloggen worden ingelezen, en de commando's die erin staan zullen worden uitgevoerd.

Voorbeeld—kijk naar dit `.bash_profile` bestand:

```
# Ik ben commentaar
echo Omgeving:
printenv | less   # equivalent aan commando SET onder DOS
alias d='ls -l'   # eenvoudig te begrijpen wat een alias is
alias up='cd ..'
echo "Ik herinner je eraan dat het pad gelijk is aan "$PATH
echo "Vandaag is het `date`"  # gebruik de uitvoer van het commando 'date'
echo "Prettige dag, "$LOGNAME
# Het volgende is een "shell functie"
```

```
ctgz() # Toon de inhoud van een .tar.gz archief.
{
  for file in $*
  do
    gzip -dc ${file} | tar tf -
  done
}
# einde .profile
```

`$PATH` en `$LOGNAME`, je raadde het goed, zijn omgevingsvariabelen. Er zijn nog veel anderen om mee te experimenteren; bijvoorbeeld, LMP voor apps als `less` of `bash`.

Door deze regel in je `/etc/profile` te plaatsen zal grofweg worden voorzien in het equivalent van `PROMPT $P$G`:

```
export PS1="\w\\$ "
```

8.2 Programma initialisatiebestanden

Onder Linux kan praktisch alles naar behoeften worden aangepast. Voor de meeste programma's zijn er één of meer initialisatiebestanden waar je wat mee kunt, vaak als een `.prognaamrc` in je home-dir. De eerste die je daarvan wellicht zal willen wijzigen zijn:

- `.inputrc`: gebruikt door `bash` voor het definiëren van key bindings;
- `.xinitrc`: gebruikt door `startx` voor het initialiseren van het X Window System;
- `.fvwmrc`: in gebruik door de window manager `fvwm`.
- `.joerc, .jstarrc`: in gebruik door de editor `joe`;
- `.jedrc`: in gebruik door de editor `jed`;
- `.pinerc`: in gebruik door de mailreader `pine`;
- `.Xdefault`: in gebruik door diverse X-programma's.

Je komt deze allemaal en nog anderen vroeg of laat tegen, LMP. Misschien dat ik je kan interesseren voor de Configuratie HOWTO, `<http://www.linuxdoc.org/HOWTO/Config-HOWTO.html>`?

9 Netwerken: begrippen

Niet alleen is "Dialup Networking" onder Linux beschikbaar, maar het is ook nog eens stabieler en sneller. De naam hiervan is "PPP", het protocol dat wordt gebruikt voor het maken van een verbinding met het Internet met behulp van modems. Het enige wat je hiervoor nodig hebt is een utility waarmee kan worden gebeld en de verbinding tot stand kan worden gebracht.

Om je mail vanaf de server van je ISP op te halen, heb je een tool genaamd "email fetcher" nodig welke het POP protocol gebruikt; wanneer de mail is opgehaald, zal het lijken alsof het direct naar je Linux-box werd bezorgd. Vervolgens gebruik je een MUA (Mail User Agent) zoals `pine`, `mutt`, `elm` of een van de vele anderen om het te beheren.

Terwijl de dialer onder Windows automatisch wordt aangeroepen wanneer je een Internet applicatie start, gaat het bij Linux net andersom: je belt eerst, en vervolgens start je de applicatie op. Een ding genaamd diald voorziet in het gebruikelijke functioneren. Het installeren en configureren van dialup networking was één van de moeilijkste taken onder Linux, maar dit is niet meer zo: raadpleeg alsjeblieft de Configuratie HOWTO.

Als laatste, iets over "Network neighborhood": je kunt je Linux-box als Windows NT/9x in een lokaal netwerk met Windows computers laten verschijnen. Het magische woord is Samba: niet de levende Braziliaanse dans, maar een implementatie van het SMB-protocol voor Linux. Ga hiervoor naar `<http://samba.anu.edu.au/samba>`.

10 Iets over programmeren

10.1 Shell Scripts: .BATch bestanden

Als je gewend was .BATch bestanden aan te maken als verkorte opdrachten voor lange opdrachtregels (ik deed dit veel), kan dit doel worden bereikt door de passende alias regels in `profile` of `.bash_profile` in te voegen. Maar als je gecompliceerdere .BATch bestanden gebruikte, dan zal je de scripttaal die beschikbaar is door de shell weten te waarderen: het is zo krachtig en goed als de oude QBasic, zo niet beter. Het bevat variabelen, structuren zoals while, for, case, if... then... else, en heel veel andere mogelijkheden: het kan een goed alternatief zijn voor een "echte" programmeertaal.

Voor het schrijven van een script—het equivalent van een .BATch bestand onder DOS—hoef je slechts een standaard ASCII-bestand met de instructies te schrijven, het op te slaan, en het vervolgens uitvoerbaar te maken met het commando `chmod +x <scriptfile>`. Typ de naam ervan in, om het uit te laten voeren.

Een waarschuwing. De systeem editor wordt `vi` genoemd, en mijn ervaring is dat nieuwe gebruikers het erg moeilijk in 't gebruik vinden. Ik ga niet uitleggen hoe je het kunt gebruiken; raadpleeg alsjeblieft Matt Welsh's boek of zoek naar een tutorial op het net. Ik volsta met hier te zeggen dat:

- je een 'i' intikt en dan je tekst, om tekst in te voegen;
- je <ESC> dan 'x' intikt, om tekens te verwijderen;
- je typt <ESC> dan `:q!`, om `vi` zonder opslaan te verlaten;
- typ <ESC> dan `:wq`, om op te slaan en te verlaten.

Een goede editor voor beginners is `joe`: start het programma door `jstar` in te tikken. je krijgt dezelfde key bindings als in de DOS/Win editor. `jed` in WordStar of IDE mode is zelfs beter. Raadpleeg paragraaf 11.5 (Waar kun je applicaties vandaan halen) om erachter te komen hoe je aan deze editors kunt komen.

Het schrijven van scripts onder `bash` is zo'n veelomvattend onderwerp dat hiervoor een volledig boek nodig zou zijn, en ik zal het onderwerp hier niet verder uitdiepen. Ik geef je slechts een voorbeeld van een shellscript, van waaruit je enkele basisregels kunt halen:

```
#!/bin/sh
# sample.sh
# Ik ben commentaar
# verander de eerste regel niet, deze moet hier aanwezig zijn
echo "Dit systeem is: `uname -a`" # gebruik de uitvoer van het commando
echo "Mijn naam is $0" # ingebouwde variabelen
echo "Je gaf me de volgende $# parameters: "$*
echo "De eerste parameter is: "$1
echo -n "Wat is je naam? " ; lees je je_naam
```

```
echo merk het verschil op: "hi $je_naam" # omsloten door "
echo merk het verschil op: ’hi $je_naam’ # omsloten door ’
DIRS=0 ; FILES=0
for file in ‘ls .‘ ; do
  if [ -d ${file} ] ; then # als het bestand een directory is
    DIRS=‘expr $DIRS + 1‘  # DIRS = DIRS + 1
  elif [ -f ${file} ] ; then
    FILES=‘expr $FILES + 1‘
  fi
  case ${file} in
    *.gif|*jpg) echo "${file}: grafisch bestand" ;;
    *.txt|*.tex) echo "${file}: tekstbestand" ;;
    *.c|*.f|*.for) echo "${file}: bronbestand" ;;
    *) echo "${file}: gewoon bestand" ;;
  esac
done
echo "er zijn ${DIRS} directory’s en ${FILES} bestanden"
ls | grep "ZxY--%%WKW"
if [ $? != 0 ] ; then # exit code van laatste commando
  echo "ZxY--%%WKW niet gevonden"
fi
echo "genoeg... typ ’man bash’ als je meer info wilt."
```

10.2 C

Onder UNIX is de systeemtaal C, hou ervan of haat het. Er zijn tevens nog heel wat andere talen (Java, FORTRAN, Pascal, Lisp, Basic, Perl, awk...) beschikbaar.

Aangenomen dat je bekend bent met C, zijn hier een aantal richtlijnen voor degenen die verwend zijn met Turbo C++ of één van de DOS varianten. Linux’s C compiler wordt `gcc` genoemd en het mist alle toeters en bellen waar gewoonlijk de DOS tegenhangers van vergezeld gaan: geen IDE, on-line help, geïntegreerde debugger, enz. Het is slechts een grove commandoregel compiler, erg krachig en efficiënt. Voor het compileren van je standaard `hello.c` geef je op:

```
$ gcc hello.c
```

waarmee een uitvoerbaar bestand genaamd `a.out` zal worden aangemaakt. Geef het uitvoerbare bestand een andere naam met

```
$ gcc -o hola hello.c
```

Om een library aan het programma te koppelen, voeg je de switch -l<libname> toe. Bijvoorbeeld, om de math library te linken:

```
$ gcc -o mathprog mathprog.c -lm
```

(De `-l<libname>` switch forceert `gcc` de library `/usr/lib/lib<libname>.so` te koppelen; dus `-lm` koppelt `/usr/lib/libm.so`).

Als je programma uit verschillende bronbestanden bestaat, zul je gebruik moeten maken van de utility `make`. Laten we ervan uitgaan dat je een expression parser hebt geschreven: het bronbestand heet `parser.c` en

bestaande uit twee header bestanden, `parser.h` en `xy.h`. Dan wil je gebruik maken van de routines in `parser.c` in een programma, stel `calc.c`, welke op zijn beurt bestaat uit # `parser.h`. Wat moet je nu doen om `calc.c` te compileren?

Je zal een zogenoemde `Makefile` moeten schrijven, die de compiler laat weten wat de afhankelijkheden tussen sources en object bestanden zijn. In ons voorbeeld:

```
# Dit is Makefile, gebruikt om calc.c te compileren
# Druk waar aangegeven op de <TAB> toets!

calc: calc.o parser.o
<TAB>gcc -o calc calc.o parser.o -lm
# calc rekent op twee objectbestanden: calc.o en parser.o

calc.o: calc.c parser.h
<TAB>gcc -c calc.c
# calc.o rekent op twee bronbestanden

parser.o:  parser.c parser.h xy.h
<TAB>gcc -c parser.c
# parser.o rekent op drie source bestanden

# einde Makefile.
```

Bewaar dit bestand als `Makefile` en typ `make` om je programma te compileren; sla het alternatief op als `calc.mak` en typ `make -f calc.mak`, en natuurlijk LMP. Je kunt wat hulp krijgen over de C-functies die in man pages worden behandeld, in sectie 3, bijvoorbeeld:

```
$ man 3 printf
```

Gebruik voor het opsporen van fouten in je programma's `gdb`. `info gdb` om te leren hoe je het kunt gebruiken.

Er zijn heel veel library's beschikbaar; onder de eersten die je wellicht wilt gebruiken bevinden zich `ncurses` (tekstmode effecten), en `svgalib` (console graphics). Veel editors kunnen als IDE fungeren; `emacs` en `jed`, bijvoorbeeld, hebben als feature syntax highlighting, automatisch inspringen, enzovoort. Haal als alternatief het package `rhide` op vanaf `<ftp://metalab.unc.edu:/pub/Linux/devel/debuggers/>`. Het is een Borland IDE-kloon, en de kans bestaat dat het je bevalt.

10.3 X11 Programmeren

Als je er wat voor voelt om met X11 programmering aan de slag te gaan (zo moeilijk is het niet), bestaan er verscheidene library's waarmee het schrijven van X11 programma's een makkie is. De belangrijkste te bezoeken sites zijn die van GTK+, `<http://www.gtk.org>`, and Qt, `<http://www.troll.no>`. Gtk+ is een op C gebaseerde widget set oorspronkelijk ontworpen voor het grafische pakket De GIMP (`<http://www.gimp.org>`), en het wordt gebruikt in de Gnome omgeving. Kdeveloper is gebaseerd op de op C++ gebaseerde Qt, die door KDE wordt gebruikt. Zeer waarschijnlijk zal je één van deze library's gebruiken.

Een aantal van de beste tools voor visueel programmeren zijn Kdevelop voor Qt, `<http://www.kdevelop.org>`, en Glade voor GTK+, `<http://glade.pn.org>`. Deze pagina geeft meer informatie: `<http://www.free-soft.org/guitool/>`.

10.3.1 Multi-Platform Programmering

Zou het niet aardig zijn als je code zou kunnen schrijven die onder Linux *en* Windows naadloos compileerde met gebruik van `gcc`? Tijdens dit schrijven zijn er een aantal widget sets die een meer of minder stabiele multi-platform programmering mogelijk maken. Zover als het de stabiliteit als volledigheid echter betreft, zou ik zeggen dat de keuze is beperkt tot praktisch één: FLTK, de Fast Light Tool Kit `<http://www.fltk.org>`. Het is verbazingwekkend klein, snel en stabiel. Het heeft tevens een semi-visuele builder genaamd Fluid.

11 De resterende 1%

Veel meer dan 1%, eigenlijk...

11.1 Draaien van DOS/Windows Apps

Ja, tot op zekere hoogte kun je DOS en Windows applicaties onder Linux draaien! Er zijn twee tamelijk goede emulators: Dosemu (`<http://www.dosemu.org>`) en Wine (`<http://www.winehq.com>`). De laatste wordt bij iedere release veel beter en de lijst met uit te voeren applicaties wordt steeds groter. Zelfs Word en Excel draaien eronder!

11.2 Gebruik van tar en gzip

Onder UNIX zijn een aantal veel gebruikte applicaties voor het archiveren en comprimeren van bestanden. `tar` wordt gebruikt om archieven te maken—het is als `PKZIP` of `Winzip` maar comprimeert niet, het maakt alleen archieven. Typ voor het maken van een archief:

```
$ tar cvf <archive_name.tar> <file> [file...]
```

Voor het extraheren van bestanden uit een archief:

```
$ tar xvf <archive_name.tar> [file...]
```

Voor het tonen van de inhoud van een archief:

```
$ tar tf <archive_name.tar> | less
```

Met `compress`, welke verouderd is en niet meer gebruikt zou moeten worden, of `gzip`:

```
$ compress <file>
$ gzip <file>
```

waarmee een gecomprimeerd bestand met de extensie `.Z` wordt aangemaakt (`compress`) of `.gz` (voor `gzip`). Deze programma's kunnen slechts één bestand tegelijkertijd comprimeren. Decomprimeren:

```
$ compress -d <file.Z>
$ gzip -d <file.gz>
```

LMP.

Er zijn ook nog de utility's `unarj`, `zip` en `unzip` (PK??ZIP compatibele). Bestanden met de extensie `.tar.gz` of `.tgz` (gearchiveerd met `tar`, en vervolgens gecomprimeerd met `gzip`) zijn ook zeer algemeen in de UNIX-wereld net als .ZIP bestanden onder DOS. Zo kun je de inhoud van een `.tar.gz` archief weer laten geven:

```
$ tar ztf <file.tar.gz> | less
```

11.3 Applicaties installeren

Ten eerste: het installeren van packages is het werk van root. De meeste Linux applicaties worden als `.tar.gz` archief gedistribueerd, die vaak bestaat uit een directory met waarschijnlijk de naam `pkgname/`, waarin de bestanden en/of subdirectory's staan. Het is een goede gewoonte om deze packages vanuit de directory `/usr/local` te installeren met het commando:

```
# tar zxf <archive.tar.gz>
```

en het README of INSTALL bestand te lezen. In de meeste gevallen worden de bronbestanden van de applicatie gedistribueerd, wat inhoudt dat je het moet compileren om de uitvoerbare bestanden aan te maken; vaak is het al voldoende om `make` en daarna `make install` in te tikken. Uiteraard heb je hiervoor de `gcc` of `g++` compiler nodig.

Andere archieven moeten worden uitgepakt vanuit `/`; dit is het geval met de `.tgz` archieven van Slackware. In andere archieven bevinden zich bestanden, maar dan zonder subdirectory - pas op dat je de boel niet verknoeit! Bekijk altijd de inhoud van het archief voordat je het installeert.

De Debian en Red Hat distributies hebben hun eigen archief formaat; respectievelijk, `.deb` en `.rpm`. De laatste wint een brede acceptatie; om een `.rpm` package te installeren, tik je in:

```
# rpm -i package.rpm
```

11.4 Tips waar je niet zonder kunt

Terugscrollen: het indrukken van <SHIFT + PAG UP> (de grijze toets) maakt dat je een paar pagina's terug kunt scrollen, afhankelijk van de hoeveelheid van het videogeheugen.

Resetten van het scherm: als je met `more` of `cat` een binair bestand bekijkt, dan kan het gebeuren dat je scherm vol met rommel staat. Om dit te herstellen, tik dan blind `reset` in of de volgende reeks met tekens: `echo CTRL-V ESC c RETURN`.

Het plakken van tekst: in console, zie hieronder; in X, klik en sleep om de tekst in een `xterm` window te selecteren, druk dan de middelste muisknop in (of de twee muisknoppen samen als je een twee-knops muis hebt) om de tekst te plakken. Er is ook een `xclipboard` (helaas, alleen voor tekst); laat je niet in de war brengen door de zeer trage response.

Gebruik van de muis: als je `gpm`, een mousedriver voor de console, hebt geïnstalleerd, dan kun je klikken en slepen om tekst te selecteren, en vervolgens (op een andere plaats) de rechtermuisknop indrukken om de geselecteerde tekst te plakken. Dit werkt ook door in de ene VC tekst te selecteren en deze in een andere VC te plakken.

Meldingen van de kernel: kijk als root in `/var/adm/messages` of `/var/log/messages` om te zien wat de kernel je te melden heeft, inclusief de meldingen bij het opstarten. Het commando `dmesg` is ook handig.

11.5 Waar vind ik applicaties

Als je je afvraagt of je je oude en vertrouwde DOS/Win applicatie door een Linux applicatie kunt vervangen, raad ik je aan eens naar de belangrijkste Linux software opslagplaats te surfen: `<ftp://metalab.unc.edu/pub/Linux>`. Andere goede startplaatsen zijn de "Linux Applications and Utilities Page" `<http://www.xnet.com/~blatura/linapps.shtml>`, de "officiële" Linux page `<http://www.linux.org>`, en `<http://freshmeat.net>`.

11.6 Een aantal dingen die je niet kon doen

Linux kan verschrikkelijk veel dat onder DOS/Windows omslachtig, moeilijk of onmogelijk was. Hier is een beknopte lijst om de smaak te pakken te krijgen:

- `at` maakt het mogelijk programma's op bepaalde tijden uit te voeren:
- `awk` is een eenvoudige maar krachtige taal om gegevensbestanden te manipuleren (en niet alleen dat). Als bijvoorbeeld `data.dat` je uit meerdere velden bestaande gegevensbestand is, drukt

  ```
  $ awk '$2 ~ "abc" {print $1, "\t", $4}' data.dat
  ```

  de velden 1 en 4 van iedere regels in `data.dat` af waarvan het tweede veld de string "abc" bevat.
- `cron` is nuttig voor het periodiek uitvoeren van taken, op opgegeven datum en tijd. Typ `man 5 crontab`.
- `file` <`bestandsnaam`> vertelt je wat `bestandsnaam` voor een soort bestand is (ASCII-tekst, uitvoerbaar bestand, archief, enz.);
- `find` (zie tevens sectie 5.3 (Directory's: Commando's vertalen)) is één van de krachtigste en handigste commando's. Het wordt gebruikt om naar bestanden te zoeken die voldoen aan verscheidene karakteristieken en waarop acties kunnen worden uitgevoerd. Algemeen gebruik van `find` is:

  ```
  $ find <directory> <expressie>
  ```

  <expressie> bevat zoekcriteria en acties. Voorbeelden:

  ```
  $ find . -type l -exec ls -l {} \;
  ```

  zoekt naar alle bestanden welke symbolische koppelingen zijn en geeft aan waarnaar ze verwijzen.

  ```
  $ find / -name "*.old" -ok rm {} \;
  ```

  zoekt naar alle bestanden die overeenkomen met het patroon het verwijdert ze, waarbij eerst om bevestiging wordt gevraagd.

  ```
  $ find . -perm +111
  ```

  zoekt naar alle bestanden waarvan de permissies overeenkomen met 111 (uitvoerbaar bestand).

  ```
  $ find . -user root
  ```

  zoekt naar alle bestanden die aan root toebehoren. Heel veel mogelijkheden hier—LMP.
- `grep` zoekt naar tekstpatronen in bestanden. Bijvoorbeeld,

  ```
  $ grep -l "geology" *.tex
  ```

  toont de *.tex bestanden waarin het woord "geology" voorkomt. Een variant op grep genaamd `zgrep` werkt op gzipped bestanden. LMP;
- **reguliere expressies** zijn complex maar het is een zeer krachtige manier om zoekbewerkingen op tekst uit te voeren. Bijvoorbeeld: `^a[^a-m]X{4,}txt$` komt overeen met een regel welke begint met een 'a', gevolgd door een willekeurig teken behalve die in het interval a-m, gevolgd door 4 of meer 'X' tekens en eindigt op 'txt'. Reguliere expressies gebruik je met geavanceerde editors, `less`, en vele andere applicaties. `man grep` voor een introductie.

- `script <script_file>` dumpt de inhoud van het scherm in een scriptbestand genaamd `script_file` totdat je het commando `exit` geeft. Handig voor het opsporen van fouten;
- `sudo` maakt het gebruikers mogelijk een aantal van de taken van root uit te voeren (b.v. het formatteren en mounten van disks;LMP);
- `uname -a` geeft je info over je systeem;
- `zcat` en `zless` zijn handig voor het doorbladeren en omleiden van met gzip gecomprimeerde bestanden zonder ze te decomprimeren. Bijvoorbeeld:

```
$ zless textfile.gz
$ zcat textfile.gz | lpr
```

- De volgende commando's komen vaak van pas: `bc, cal, chsh, cmp, cut, fmt, head, hexdump, nl, passwd, printf, sort, split, strings, tac, tail, tee, touch, uniq, w, wall, wc, whereis, write, xargs, znew.` LMP.

11.7 UNIX oefenen onder DOS/Win

Geloof het of niet, er zijn prima tools die voorzien in een op UNIX lijkende omgeving onder DOS/Windows! Een daarvan is de Djgpp suite (<http://www.delorie.com/djgpp/>) voor DOS, en Cygwin (<http://www.cygnus.com/cygwin>) is een complexere port voor Win32. In beiden zijn dezelfde GNU ontwikkelaars tools en utility's als Linux opgenomen; je krijgt echter niet dezelfde stabiliteit en performance.

Probeer Djgpp als je de smaak van Linux te pakken wilt krijgen. Download en installeer de volgende bestanden (tijdens dit schrijven, is de laatste versie 2.0.2): `djdev202.zip, bnu281b.zip, bsh1147b.zip, fil316b.zip, find41b.zip, grep22b.zip, gwk303b.zip, lss332b.zip, shl112b.zip.`. In installatie instructies is voorzien, en je kunt assistentie vinden op <news:comp.os.msdos.djgpp>.

In het bijzonder is het gebruik van `bash` onder DOSWin een opluchting. Om het op je juiste wijze te configureren, wijzig je het meegeleverde bestand `BOOT.BAT` om je installatie weer te geven, plaats deze bestanden dan in je home-directory (in de Windows partitie) in plaats van degene waarin is voorzien:

```
# dit is _bashrc

LS_OPTIONS="-F -s --color=yes"
alias cp='cp -i'
alias d='ls -l'
alias l=less
alias ls="ls $LS_OPTIONS"
alias mv='mv -i'
alias rm='rm -i'
alias u='cd ..'

# dit is _bprof
if [ -f ~/_bashrc ]; then
  . ~/_bashrc
fi
PS1='\w\$ '
PS2='> '
CDPATH="$CDPATH:~"
# stuff for less(1)
LESS="-M-Q"                     # lange prompt, stil
LESSEDIT="%E ?lt+%lt. %f"       # wijzig eerste regel
```

```
VISUAL="jed"                    # editor
LESSCHARSET=latin1              # toon geaccentueerde letters
export PS1 PS2 CDPATH LS_OPTIONS LESS LESSEDIT LESSOPEN VISUAL LESSCHARSET
```

11.8 Algemene extensies en daaraan gerelateerde programma's

Je komt mogelijk heel wat soorten bestandsextensies tegen. Uitgezonderd de meer exotische extensies (d.w.z. fonts, enz.), tref je hier een lijst van aan:

- `1 ...` 8: man pages. Lees ze met `groff -Tascii -man` <`file.1`>.
- `arj`: archief aangemaakt met `arj`.
- `dvi`: uitvoerbestand geproduceerd met TeX (zie hieronder). `xdvi` om het te bekijken; `dvips` om het om te zetten in een PostScript `.ps` bestand.
- `gz`: archief gemaakt met `gzip`.
- `info`: info bestand (soort alternatief van man pages). Haal hier `info` voor op.
- `lsm`: Linux Software Map bestand. Het is een gewoon ASCII bestand met de beschrijving van een package.
- `ps`: PostScript bestand. Haal voor het bekijken of afdrukken `gs` op en optioneel `ghostview` of `gv`.
- `rpm`: Red Hat package. Je kunt het op ieder systeem installeren met de package manager `rpm`.
- `taz, tar.Z`: archief aangemaakt met `tar` en gecomprimeerd met `compress`.
- `tgz, tar.gz`: archief gemaakt met `tar` en gecomprimeerd met `gzip`.
- `tex`: tekstbestand voor TeX, een krachtige zetsysteem. Het package `tex` is in diverse distibuties beschikbaar.
- `texi`: texinfo bestand, kan zowel TeX als info bestanden produceren (cp. `info`). Haal `texinfo` op.
- `xbm, xpm, xwd`: grafisch bestand.
- `Z`: archief aangemaakt met `compress`.

11.9 Bestanden converteren

Als je tekstbestanden tussen DOS/Win en Linux uit moet wisselen, wees je dan bewust van het "einde regel" probleem. Onder DOS eindigt iedere regel tekst op een CR/LF (dat is een ASCII 13 + ASCII 10), onder Linux op een LF. Als je een DOS tekstbestand onder Linux wijzigt, zal iedere regel zeer waarschijnlijk op een vreemd–uitziend 'M' teken eindigen; een Linux tekstbestand zal onder DOS verschijnen als een kilometers lange enkele regel zonder alinea's. Er zijn een aantal tools, `dos2unix` en `unix2dos`, om de bestanden te converteren.

Als je bestanden bestaande uit alleen tekst geaccentueerde tekens bevat, zorg er dan voor dat ze onder Windows (met Notepad/Kladblok) zijn aangemaakt en niet onder gewoon DOS; anders zullen alle geaccentueerde tekens zijn verknald.

11.10 Vrij verkrijgbare Office Suites

Ja, je kunt voor niks krijgen wat anders heel veel geld zou kosten!

StarOffice (<http://www.sun.com/staroffice>.) is thans de enige keuze, alhoewel Koffice in de pijplijn zit (<http://www.koffice.org>). StarOffice is groot en traag, maar hoe dan ook erg goed; het biedt heel veel functionaliteit die niet in Microsoft Office is te vinden. Het kan bovendien Word en Excel bestanden inlezen, alhoewel de conversie niet altijd perfect is.

Een ander goed package is Corel WordPerfect, een vrij verkrijgbare uitgave die voor download beschikbaar is. Moet ik nog meer zeggen? Haal het op: <http://www.corel.com>.

12 Voor het moment, het einde

Gefeliciteerd! Je hebt nu een klein beetje van UNIX begrepen en bent zover om ermee beginnen te werken. Denk eraan dat je kennis van het systeem nog steeds beperkt is en dat ervan je wordt verwacht meer met Linux te oefenen om het comfortabel te kunnen gebruiken. Maar als het enige dat je moest doen een boel applicaties te verkrijgen en ermee beginnen te werken, dan is alles wat ik hierin heb opgenomen voldoende.

Ik ben er zeker van dat je Linux met plezier zal gaan gebruiken en er meer over zal gaan leren – iedereen doet dit. Ik wed ook dat je nooit meer naar DOS/Win terug zal keren! Ik hoop dat ik duidelijk ben geweest en de 3 of 4 lezers een goede dienst heb bewezen.

12.1 Copyright

Raadpleeg alsjeblieft de Linux Documentatie Project homepage, <http://www.linuxdoc.org>, als je vragen hebt.

12.2 Disclaimer

Dit document wordt zoals het is geleverd. Ik heb er veel moeite voor gedaan het zo accuraat mogelijk te schrijven, maar je gebruikt de informatie hierin op eigen risico. In geen geval zal ik verantwoordelijk kunnen worden gesteld voor enige schade voortvloeiend uit het gebruik van dit werk.

Veel dank aan Matt Welsh, de auteur van “Linux Installation and Getting Started”, aan Ian Jackson, de auteur van “Linux frequently asked questions with answers”, aan Giuseppe Zanetti, de auteur van het boek “Linux”, aan alle mensen die me aanbevelingen mailden, en vooral aan Linus Torvalds en GNU die ons Linux gaven.

Feedback is welkom. Neem gerust contact met me op voor verzoeken, aanbevelingen, flames, enz.

Veel plezier met Linux en in ’t leven,

Guido =8-)