Een nieuwe groep maken gaat met **addgroup <nieuwe-groepnaam>**. Een bestaande groep 'weggooien' is over het algemeen geen goed idee.

Instellingen voor adduser en addgroup staan in **/etc/adduser.conf**.

13.2.4 Eigenaar en groep van bestanden

Bestanden hebben één eigenaar en één groep, waarvoor permissies kunnen worden opgegeven, zie paragraaf 4.11.

Als root kun je de eigenaar en/of de groep van een bestand veranderen met het commando **chown <username>:<groupname> <file(s)>**, waarbij je **<username>** of **:<groupname>** ook mag weglaten. Met **chgrp <groupname> <file(s)>** kan alleen de groep van een bestand worden veranderd.

Als je als gewone gebruiker lid bent van een aantal groepen, kun je zelf ook **chgrp** gebruiken om een bestand van groep te veranderen, maar alleen 'jouw' groepen mogen dan gebruikt worden.

Als je van plan bent veel bestanden te gaan maken die een andere groep moeten krijgen dan jouw primaire groep, kun je met **sg <groupname>** tijdelijk een andere primaire groep krijgen. Dit moet natuurlijk wel weer een groep zijn waar jij lid van bent.

13.2.5 Password vergeten

Het kan gebeuren dat je als gewone gebruiker je password vergeet. De procedure is eenvoudig: geef als root het commando **passwd <jouwusername>** en verzin een beter-te-onthouden-password.

Veel ernstiger is het, als je het password van root vergeten bent, maar ook dat kan opgelost worden. Je moet het systeem met Ctrl-Alt-Del rebooten en dan opstarten van een rescue disk (voor het maken daarvan, zie paragraaf 3.1.4), en ga in het installatie-hoofdmenu direct met LinkseAlt-F2 naar de tweede virtuele console. Geef daar een Enter en dan een commando als **mount /dev/hda5 /target** om je root partitie te mounten. Dan kun je met **ae /target/etc/passwd** of **ae /target/etc/shadow** de passwords editen. Je moet van de regel 'root:<passwd>:<getal>:<getal>:...' het tweede veld helemaal leeghalen, dan wordt het dus 'root::<getal>:<getal>:...'. Geef dan de commando's **umount /target** en **reboot** (vergeet niet de rescue disk uit de floppy drive te halen). Zodra je Linux weer normaal opgestart hebt, log je in als root, daar heb je nu geen password meer voor nodig. Met **passwd** geef je root vervolgens een nieuw password.

Meer informatie over alle genoemde commando's staat in hun respectievelijke manual pages.

13.3 Grote geheugens

Als je meer dan 64 MB werkgeheugen hebt, kan het gebeuren dat Linux er maar 64 detecteert, meestal vanwege een 'verouderd' BIOS (sommige Compaq's geven zelfs maar 16 MB door). Als je toch meer geheugen wilt gebruiken, moet je dat zelf bij het opstarten aan de kernel doorgeven.

Bij het gebruik van de Custom Boot floppy, moet je het bestand **syslinux.cfg** op die diskette wijzigen. Dat kan in DOS, maar ook in Linux (met **mcopy** even naar bijvoorbeeld **/tmp** kopiëren). Achter de regel "APPEND root=/dev/hdXX ro" voeg je toe "mem=128MB" of iets dergelijks.

Bij LOADLIN vanuit DOS/Windows moet je bijvoorbeeld "mem=128MB" toevoegen aan de LOADLIN command-line (waar al "root=/dev/hdXX" staat).

LILO-gebruikers moeten **/etc/lilo.conf** aanpassen. Voor de eerste "image=/XX"-regel moet een regel "append="mem=128MB"" komen (de buitenste aanhalingstekens niet overnemen, de binnenste wel). Als er al een "append"-regel bestaat, moet je "mem=128MB" eraan toevoegen. Vergeet niet **/sbin/lilo** te draaien als je **/etc/lilo.conf** veranderd hebt!

Let op: sommige computers bieden niet het totale geheugen ter gebruik aan. Om problemen te voorkomen, kan het verstandig zijn 1 MB te weinig aan de Linux kernel door te geven.

1. Inleiding

Er is iets grondig aan het veranderen in de software-wereld. Waar vroeger uitsluitend grote commerciële instanties de totale markt in handen hadden, is er nu een andere ontwikkeling zichtbaar. Steeds meer mensen en bedrijven willen af van monopolisten en gebruiken 'vrije' software. 'Vrije' software mag je gebruiken zonder ervoor te betalen. Je krijgt zelfs de broncode erbij, die je naar believen mag aanpassen en verder distribueren.

Linux is ongetwijfeld het grootste vrije-software project tot nu toe. Een compleet, gratis en vooral goed operating system. Origineel voor PC's vanaf de '386, nu ook voor Alpha's, Sparcs, Amiga's, (Power)Macs, NetWinders, PalmPilots en diverse andere. Gemaakt door honderden vrijwilligers, stuk voor stuk software-experts.

Debian GNU/Linux biedt naast het complete Linux operating system een enorm aantal programma's voor elke denkbare toepassing -- allemaal vrije software. Alle programma's zijn ondergebracht in uniforme pakketjes, die gemakkelijk geïnstalleerd en verwijderd kunnen worden. Debian wordt net als Linux verzorgd door honderden vrijwilligers.

Deze handleiding geeft een complete introductie in Debian GNU/Linux. Als eerste wordt de noodzakelijke basiskennis over onder andere Linux behandeld. Dan volgt een uitvoerige beschrijving van de installatie van Debian. Tenslotte wordt de 'Linux-newbie' op een zelfstudie-achtige manier ingeleid in de basis-bezigheden op een Debian systeem.

Debian GNU/Linux draait op elke IBM-compatible PC, vanaf de 80386. Er is minimaal 4 MB intern geheugen nodig en 65 MB harde-schijf-ruimte; maar let op: tijdens de installatie moet er tenminste 12 MB intern geheugen beschikbaar zijn (wanneer nodig kun je misschien ergens extra geheugen lenen, of de harde schijf tijdelijk in een andere computer hangen). Voor tekstmode kan elke videokaart/monitor gebruikt worden; voor grafische mode is minimaal een VGA-kaart en -monitor nodig. Aanbevolen voor "lekker" werken: minstens Pentium (of kloon), minstens 16 MB werkgeheugen en minstens 500 MB harde-schijf-ruimte. Maar als je niet al te veel eisen stelt, werkt een '486 met 8 MB ook nog heel aardig. Als je wilt 'upgraden', is het werkgeheugen het eerste dat aandacht verdient -- uitbreiding tot zo'n 32 MB geeft meer snelheidswinst dan je met een snellere processor zou krijgen. Dat komt, omdat Linux al het niet-gebruikte werkgeheugen als disk-cache reserveert, wat schijftoegang enorm kan versnellen.

Iets over de typografie: als er tekst **op deze manier** is aangegeven, gaat het om een programma-naam of een commando wat je kunt intypen. Om papier (en dus kosten) te besparen, staan de commando's niet op aparte regels, maar in de lopende tekst.

```
Tekst die in een bestand staat
(of in een bestand
gezet moet worden),
wordt op deze manier aangegeven.
```

Dan nog iets over de uitspraak van Debian. Die is namelijk "dèbbiejun" en niet "díébiejun" zoals het op het eerste gezicht lijkt. De naam Debian is een samentrekking van Debra en Ian Murdock, die het Debian-project zijn gestart.

Tenslotte nog dit: lees deze handleiding van voor tot achter door, en **sla niets over**. Als je zomaar hier en daar wat leest, krijg je gegarandeerd een systeem dat maar half werkt en moet je dus opnieuw beginnen. Sommige stukjes zijn aan de lastige kant, en die zul je twee of meer keer moeten lezen. In het algemeen: lees eerst een hele paragraaf door, voordat je gaat doen wat erin staat. Als je hulp nodig hebt, kijk dan in paragraaf 4.21.

13. Andere interessante zaken

In dit hoofdstuk nog enkele zaken die niet echt berekking hebben op de voorgaande hoofdstukken, maar toch van belang kunnen zijn.

13.1 De datum en de tijd

In Linux bekijk je de huidige datum en tijd met het commando **`date`**, zie ook de manual page. Maar daarmee is nog lang niet alles gezegd.

In een computer zitten twee klokken, een hardware-klok en een software-klok. De hardware-klok werkt altijd, of de computer nu aan of uit staat, en valt in dezelfde categorie als het CMOS Setup-geheugen. De hardware-klok is ook met de CMOS Setup (meestal DEL tijdens het opstarten) te veranderen. De software-klok werkt alleen als er op de computer een programma draait dat die klok bijhoudt. Dat programma moet dus op geregelde tijden ergens een (seconde-)teller ophogen.

De Linux kernel heeft zo'n software-klok, die de 'system time' bijhoudt. Tijdens het opstarten van de kernel wordt de software-klok ingesteld op de waarde die op dat moment van de hardware-klok (ook wel 'CMOS clock' genoemd) af te lezen valt. Vanaf dat moment wordt de hardware-klok 'vergeten' en wordt alleen naar de software-klok gekeken. Dat is omdat de Linux software-klok vele malen nauwkeuriger is dan de hardware-klok. Met **`date`** zie je dus de software-klok.

13.1.1 De klok goedzetten

Je kunt de software-klok (als root) verzetten met een commando als **`date --set '04/02/01 19:55:30'`**, dan wordt het 2 april 2001, 4-en-een-halve minuut voor 8 uur 's avonds.

De hardware-klok krijgt automatisch de tijd van de software-klok als je Linux afsluit. Dat is omdat de software-klok meestal nauwkeuriger loopt dan de hardware-klok. Een extraatje is dat Linux de hardware-klok automatisch corrigeert als die te snel of te langzaam lijkt te lopen. In de praktijk blijkt dat echter niet altijd even goed te werken. Het is daarom in het algemeen een goed idee om, steeds als je **`date --set`** hebt gedaan, ook **`echo '' > /etc/adjtime`** te doen. Daarmee wordt de 'oude' informatie vergeten en wordt de tijd meestal nauwkeuriger bijgehouden.

Met de hardware-klok hoef je dus zelf nooit iets te doen, dat gaat automatisch. Meer informatie over hardware- en software-klok staat in **`man hwclock`**.

13.1.2 Tijdzones, zomertijd en wintertijd

De Europese tijd (tijdzone CEST, Central European Standard Time) loopt in de winter één en in de zomer twee uur voor op GMT (Greenwich Mean Time, of UTC, Coordinated Universal Time). Het voordeel van GMT is, dat het geen zomer- en wintertijd kent, en dus nooit een uur vooruit of terug gezet hoeft te worden.

Als de hardware-klok op GMT draait (en als Linux dat weet), zal Linux je nog steeds de lokale tijd (CEST dus) geven. Maar DOS bijvoorbeeld weet niets van tijdzones af en gaat ervan uit dat de hardware-klok op lokale tijd draait. Als je dus naast Linux nog een ander operating system draait, kun je de hardware-klok het beste op lokale tijd laten draaien.

Van elke tijdzone (eigenlijk van elk land) is een aantal zaken bekend, in de directory **`/usr/share/zoneinfo`**. Eén van die dingen is de zomer- en wintertijd. Als de hardware-klok op GMT draait, zul je merken dat de zomer- en wintertijd automatisch wordt aangepast (in de software-klok dus, de GMT van de hardware-klok kent immers geen zomer- en wintertijd). Als de hardware-klok op lokale tijd draait, gaat het soms wel, en soms ook niet goed; ook als je af en toe Windows 95/98 draait, die zelf ook zomer- en wintertijd kan omschakelen. Het kan nooit kwaad om op dergelijke momenten te kijken of alles nog goed loopt, en het zonodig aan te passen (en vooral een **`echo '' > /etc/adjtime`** niet vergeten, zie boven).

Voor gebruikers uit een ander land of een andere tijdzone is er nog een leuk trucje: zet de environment variabele TZ (van timezone) op de juiste tijdzone, en je krijgt automatisch de omgerekende tijd. Probeer maar eens **`export TZ=US/Central`** en dan **`date`**, of **`ls -alF`**. Type **`unset TZ`** en bekijk hoe de datum of de directory er werkelijk uitziet.

Harddisk
Op de *Harddisk* (NL: *harde schijf*) staan programma-bestanden en gegevens-bestanden. Er zijn twee typen (lees: aansluitingsvormen), *IDE* en *SCSI*. IDE's zijn het goedkoopst en kunnen door elke Multi-I/O kaart (eventueel op het Main Board) worden aangestuurd; SCSI versies zijn soms sneller en hebben een aparte *SCSI Controller* (insteekkaart) nodig. Harde schijven van vele GB's (gigabytes, 'miljard' geheugenplaatsen; 1 GB = 1024 MB = 1073741824 bytes) zijn leverbaar.

Andere apparaten
Zowel binnen als buiten de computer'kast' zijn nog een aantal andere apparaten. Te denken valt aan een *geluidskaart* (*sound card*), een *netwerkkaart* (*network adapter*), een *modem*, de *CDROM-speler* (*CDROM drive*), het *toetsenbord* (*keyboard*), de *muis* (*mouse*), de *monitor* en de *printer*. Een aantal hiervan zal in het vervolg nog ter sprake komen.

2.2 Programma's op de computer, het operating system, de kernel

Op een computer draaien *programma's* (dat is de bedoeling tenminste). Een programma is niets meer dan een hele lange rij van allemaal kleine opdrachtjes, zoals 'verhoog de inhoud van geheugenplaats 6314 met 3', 'kijk of het groter is dan nul', 'spring 41 opdrachten terug'. Al die duizenden kleine opdrachtjes bij elkaar vormen bijvoorbeeld een tekstverwerker.

Programma's staan meestal in (programma-)bestanden op de harde schijf. Maar om ze uit te kunnen voeren, moeten ze in het werkgeheugen staan. Op de één of andere manier moet het programma dus van de harde schijf naar het werkgeheugen worden verplaatst. Dat noemen we het *laden* (*loading*) van het programma.

Het laden en starten van bijvoorbeeld een tekstverwerker wordt geregeld door een *operating system*. Dat is een collectie van programma's die op zichzelf niet zo nuttig is, maar wel onmisbaar. Het *operating system* laadt een programma, wijst er voldoende werkgeheugen aan toe voor gegevensopslag, geeft programma's toegang tot bijvoorbeeld documenten op de harde schijf, en ruimt alles netjes op als het programma afgelopen is. Het levert daartoe een hele reeks van programma-delen, of *routines*, die door andere programma's kunnen worden gebruikt (aangeroepen). Bijvoorbeeld 'open bestand XYZ', 'lees het volgende blok gegevens uit dat bestand', 'laad en start nu programma ABC', maar ook 'zet een j op die-en-die plaats op het scherm'. Deze routines maken deel uit van de *kern* van het operating system, in het Engels: *kernel*.

Omdat het operating system natuurlijk niet weet welk programma je wilt gaan gebruiken, wordt als eerste een standaard-programma gestart, waarin je kunt aangeven welk programma je wilt starten. Zo'n programma presenteert bijvoorbeeld een *prompt* waarachter je de naam van het te starten programma(-bestand) moet intypen. Dat eerste 'doorverwijs-programma' noemen we een *shell*. De shell hoort bij het operating system, maar niet bij de kernel, het is een apart programma. Mogelijkheden om bijvoorbeeld bestanden en directory's te bekijken, kopiëren en verplaatsen horen ook bij het operating system; ze kunnen als interne commando's in de shell beschikbaar zijn, of als aparte programma's.

Laten we eens een bekend voorbeeld analyseren, zodat we een beter idee krijgen waar we het over hebben.

Het bekendste operating system is ongetwijfeld MSDOS (Microsoft Disk operating system). De kernel hiervan bestaat uit de bestanden **`MSDOS.SYS`** en **`IO.SYS`**. Deze bestanden worden als eerste geladen als de computer wordt aangezet (niet helemaal waar, eerst de bootsector nog -- daar gaan we het nog over hebben). Als de MSDOS kernel is gestart, wordt eerst **`CONFIG.SYS`** gelezen, vervolgens wordt de shell **`COMMAND.COM`** gestart, die op zijn beurt de opdrachten uit **`AUTOEXEC.BAT`** gaat uitvoeren. Als dat alles achter de rug is, komt de DOS-prompt op het scherm en kunnen er programma's worden gestart. **`COMMAND.COM`** is trouwens niet de enige shell voor MSDOS, Norton heeft bijvoorbeeld **`NDOS.EXE`**. Het commando '**`COPY A B`**' is in **`COMMAND.COM`** ingebakken; **`XCOPY`**, dat ietsje uitgebreider is, is een apart programma. Beiden behoren tot het operating system.

In DOS kan maar één programma tegelijk draaien: zolang de tekstverwerker actief is, kun je niets meer met **`COMMAND.COM`** (of een ander programma) doen. In Windows-varianten draaien een aantal programma's tegelijkertijd, dat heet *multitasking*, letterlijk 'meerdere taken'. Je kunt die programma's 'door elkaar' gebruiken, zonder ze steeds af te hoeven sluiten.

2. Achtergronden en basiskennis van Linux en Debian

Om verstandig te kunnen omgaan met een willekeurig ding, is het noodzakelijk er het één en ander vanaf te weten. Dat geldt zeker bij de installatie en het gebruik van Debian GNU/Linux (kortweg 'Debian'). Ook als je (bijvoorbeeld op het Internet) een vraag stelt of deelneemt aan een discussie, is het beter dat je niet compleet 'ongeletterd' voor de dag komt. Vandaar dat ik hier wat basiskennis wil aanreiken. Omdat deze tekst niet alleen is bedoeld voor hoogbegaafden, zal ik van onderaf beginnen. (Dat wil niet zeggen dat geleerde heren en dames het onderstaande kunnen overslaan -- correcties en aanvullingen zijn altijd welkom!)

2.1 De computer

Zoals gezegd, beginnen we nogal heel erg 'basic'. Noodzakelijk voor beginnende computeraars en ook nuttig voor gevorderden, omdat terloops een aantal (ongebruikelijke) Engelse termen aan de orde komen. Zoals je misschien al wel door hebt, is Linux (en daarmee Debian) vrijwel compleet Engels en een aantal vaktermen kunnen je raadselachtig voorkomen. Een beetje Engels behoort ook bij de basiskennis en aangezien Microsoft alles maar naar 'goed Nederlands' vertaalt, kan ik er niet vanuit gaan dat die kennis aanwezig is.

Het gaat er niet om, dat je alle termen begrijpt, het is meer dat je weet in welke richting je moet zoeken als je een term leest. Kom je er zelf niet uit, dan kun je bijvoorbeeld terecht in de nieuwsgroep nl.comp.hardware (zie ook paragraaf 4.21).

CPU
De *CPU*, afkorting van *Central Processing Unit* (NL: *Centrale Verwerkings Eenheid*, *CVE*), ook *Processor*, is het hart van de computer. Door deze grote computerchip worden de programma's uitgevoerd. Processors hebben een bepaald merk en type, zoals *Intel 80486* ("een 486"), *Intel Pentium* (of *80586*) of *AMD K6*. De klokfrequentie bepaalt de snelheid van de computer, bijvoorbeeld 233 MHz (megahertz, miljoen cycli per seconde).

Main Board
De CPU is gemonteerd op het *Main Board* (afgekort: *MB*, of *MoBo*, NL: *Moederbord* of ook *Moederkaart*). Dat is een grote plaat (printplaat, 'PCB') die allerlei chips, geheugen en andere apparaten met de CPU verbindt.

Werkgeheugen
Het programma dat wordt uitgevoerd en de gegevens die het nodig heeft, staan in het (*werk-*)*geheugen* of *Main Memory*, ook aangeduid met *RAM* (*Random Access Memory*, willekeurig toegankelijk geheugen, lees- en schrijfbaar). Tegenwoordig is 32 MB (megabyte, 'miljoen' *bytes* of geheugenplaatsen; 1 MB = 1024 x 1024 = 1048576 bytes) zo'n beetje standaard. Termen die met de uitvoering van de geheugenchips te maken hebben zijn bijvoorbeeld *SIMM*, *DIMM*, *SDRAM*, *EDO RAM*.

Insteekkaarten
Het Main Board biedt plaats aan diverse insteekkaarten, zoals een videokaart, multi-I/O kaart, geluidskaart, netwerkkaart en SCSI kaart. Er is een trend gaande om steeds meer van die uitbreidingskaarten in het Main Board te integreren.

Videokaart
De *videokaart* of *Video Card* zorgt ervoor dat de *monitor* de juiste signalen krijgt, zodat er bijvoorbeeld letters of plaatjes op het scherm verschijnen. In de loop van de tijd zijn diverse standaarden ontwikkeld, die steeds meer konden, zoals *Hercules*, *CGA*, *EGA*, *VGA* en *SVGA* (*Super VGA*). In de grafische mode moet voor elk beeldpunt (*pixel*) worden onthouden welke kleur het heeft. Dat kost best wat geheugen, daarom heeft elke nieuwe videokaart minimaal 1 MB geheugen (*Video RAM*). De videokaart en monitor bepalen samen welke *resoluties* mogelijk zijn, bijvoorbeeld 640 bij 480 beeldpunten of 1280 bij 1024. De hoeveelheid geheugen van de videokaart bepaalt hoeveel kleuren er per beeldpunt mogelijk zijn, bij een gegeven resolutie. Videokaarten die op het Main Board zijn geïntegreerd, snoepen vaak 1 tot 2 MB af van het werkgeheugen.

Multi-I/O Controller
De *Multi-I/O Controller* regelt een heleboel *I/O* (*Input* en *Output*, invoer en uitvoer). Zoals daar zijn: *harddisk* (*harde schijf*), *floppy disk drive* (*diskettes*, "schijfjes"), een *parallelle poort* (*Parallel Port*) voor aansturing van de printer, en meestal twee *seriële poorten* (*Serial Ports*) voor bijvoorbeeld een *muis* en een *modem*. Tegenwoordig is de I/O Controller meestal op het Main Board geïntegreerd.

Als root kun je de standaard (system-wide) tijdzone instellen met het programma **`/usr/sbin/tzconfig`**; in /etc/default/rcS kun je aangeven of de hardware-klok op GMT (=UTC) loopt.

13.2 Gebruikers, groepen en passwords: /etc/passwd en /etc/group

Gegevens over gebruikers worden opgeslagen in **`/etc/passwd`**. Dat is een database-achtig tekstbestand: elke gebruiker heeft één regel en elke regel heeft een aantal *velden* die gescheiden worden door dubbele punten (':'). Die velden zijn in volgorde: gebruikersnaam, password, gebruikersnummer (*uid*, *user id*), 'primair' groepsnummer (*gid*, *group id*), 'informatie', home directory, opstartshell. Het veld 'informatie' is ook weer onderverdeeld in komma-gescheiden subvelden, namelijk volledige naam, kamernummer, werk-telefoonnummer, thuis-telefoonnummer, 'andere informatie'.

13.2.1 Passwords

Het password dat in **`/etc/passwd`** staat is *ge-encrypt* ('gecodeerd'). Dit encrypten is een één-richtings-actie, dat wil zeggen dat het echte password wel in de ge-encrypte code kan worden omgezet, maar niet andersom. Tijdens het inloggen wordt het door jou ingetypte password ook ge-encrypt en wordt vergeleken of de ge-encrypte codes hetzelfde zijn.

Het is in principe een veiligheidsrisico als het ge-encrypte password in **`/etc/passwd`** staat. Dat bestand is namelijk publiekelijk toegankelijk en er zijn programma's beschikbaar die hele lijsten met veelgebruikte passwords gaan 'uitproberen'. Daarom is er *shadowing* bedacht: de passwords staan nu in een apart bestand **`/etc/shadow`**, waar alleen bepaalde programma's (zoals **`login`** en **`passwd`**) bij kunnen; in **`/etc/passwd`** staan x-jes op de plaats van het ge-encrypte password. Je kunt shadowing als root aan- en uitzetten met **`shadowconfig on`** of **`off`**; de bestanden **`/etc/passwd`** en **`/etc/shadow`** worden daarmee automatisch aangepast.

Je kunt je password veranderen met het commando **`passwd`**, dat dan vraagt om je oude en (twee keer) je nieuwe password. Als je root bent, kun je het password van een willekeurige gebruiker veranderen met **`passwd <username>`**.

Je kunt de inhoud van het 'informatieveld' veranderen met het commando **`chfn`** (of als root eventueel **`chfn <username>`**) en de opstartshell met **`chsh`**. NB: verander nooit de opstartshell van root!

13.2.2 Groepen

Iedere gebruiker is lid van precies één '*primaire*' *groep*. Bestanden die door die gebruiker worden aangemaakt, worden van die groep (bekijk dat met **`ls -alF`**). Maar naast de primaire groep kunnen gebruikers ook lid zijn van andere groepen, dat wil zeggen dat ze dan ook bestanden kunnen bekijken/veranderen die leesbaar/schrijfbaar zijn voor zo'n andere groep.

De groepen worden bijgehouden in **`/etc/group`**. In het vierde veld (achter de laatste dubbele punt) staan de komma-gescheiden namen van gebruikers die lid zijn van die groep. Het lidmaatschap van de primaire groep wordt echter niet in **`/etc/group`** aangegeven, maar in het vierde veld van **`/etc/passwd`**. In **`/etc/group`** worden ook de groepnamen gekoppeld aan de groepnummers (die bijvoorbeeld in dat vierde veld van **`/etc/passwd`** gebruikt worden).

Je kunt met het commando **`groups`** zien, van welke groepen je lid bent; het commando **`id`** geeft nog wat meer informatie. Merk op dat groeps-veranderingen pas merkbaar zijn als je een volgende keer inlogt.

13.2.3 Nieuwe gebruikers en groepen

Zoals in paragraaf 5.4 al is aangegeven, maak je een nieuwe gebruiker aan met **`adduser <nieuwe-username>`**. Standaard krijgt die gebruiker een eigen nieuwe (primaire) groep, met zijn eigen naam. Je kunt nieuwe gebruikers ook direct aan een andere, bestaande groep toevoegen (d.w.z. een andere primaire groep geven) door **`adduser --ingroup <groepnaam> <nieuwe-username>`**.

Een bestaande gebruiker toevoegen aan een bestaande groep gaat met **`adduser <username> <groepnaam>`**. Gebruikers uit een groep verwijderen die niet hun primaire groep is, kan alleen door met de hand **`/etc/group`** te editen. Gebruikers een andere primaire groep geven, kan alleen door het juiste veld in **`/etc/passwd`** te wijzigen.

Om een bestaande gebruiker 'weg te gooien', gebruik je **`userdel -r <username>`**. Hiermee wordt zonder bevestiging ook de home directory gewist, dus voorzichtig!

2.3 Het operating system met de naam Linux

Het operating system (vaak afgekort tot *OS*) waar we het in het vervolg over gaan hebben, heet *Linux*. Hoewel, dat is niet helemaal juist. De *kernel* van het operating system heet Linux, maar in de volksmond wordt dat vaak door elkaar gehaald. De officiële naam is GNU/Linux. Dat komt omdat praktisch alle *utility* programma's ('gereedschap') voor bijvoorbeeld kopiëren en verplaatsen, maar ook de veel gebruikte C en C++ compiler, van het GNU project komen. De *GNU utility's* en de *Linux kernel* vormen *samen* een operating system, vandaar GNU/Linux.

Een beetje geschiedenis
Om een lang verhaal kort te houden: Linux is een *UNIX*-achtige kernel waar de Finse informatica-student Linus Torvalds aan is begonnen en dat onder zijn leiding van de grond af is opgebouwd door honderden vrijwilligers.

UNIX is een OS dat vanaf de eerste computers (zogenaamde *mainframes*) is gebruikt, en dat als voorbeeld diende voor de diverse DOS-versies van vandaag. UNIX staat bekend om zijn zeer flexibele multitasking, maar ook omdat het een *multi-user* OS is. Dat wil zeggen dat er veel mensen tegelijkertijd op één enkele computer met één enkele CPU kunnen werken. In vroeger dagen was dat noodzakelijk, omdat één computer eigenlijk al onbetaalbaar was. Deze multi-user mogelijkheden werden benut door 'gewoon' een aantal extra beeldschermen en toetsenborden op dezelfde computer aan te sluiten. Een combinatie van monitor en toetsenbord wordt een *terminal* genoemd. Zo'n terminal zit met een seriële kabel aan de computer vast, op dezelfde connector als bijvoorbeeld een modem. De terminal geeft aan de computer door welke toetsen er worden ingedrukt, en de computer vertelt de terminal welke letters er op de monitor moeten verschijnen. De monitor en het toetsenbord die aan de computer vastzitten op de 'normale' manier, worden geen terminal, maar *console* genoemd. Linux heeft dezelfde mogelijkheden -- en meer zelfs.

Linus heeft alleenmaar de kernel 'geschreven'. Maar een kernel op zich kan nog niets, daar zijn utility's, een shell, en andere programma's voor nodig. Nu wil het geval, dat net toen Linux iets begon te worden, de GNU utility's beschikbaar kwamen. De GNU utility's zijn nieuw geprogrammeerde versies van alle standaard UNIX commando's / utility's (inclusief een shell), en zijn vrij gebruikbaar, ze komen zelfs met volledige programmacode (*source code*). Omdat ook Linux 'vrij' is, lag het voor de hand om de GNU utility's te gebruiken bij de Linux kernel. Aldus is GNU/Linux ontstaan.

Distributies
Naast de Linux kernel en de GNU utility's zijn veel andere programma's geschreven voor Linux. Net als de kernel en de GNU utility's, worden de meeste Linux programma's uitsluitend in de oorspronkelijke (niet-uitvoerbare) programmacode geleverd (meestal C of C++). Gebruikers moesten deze programmacode zelf omzetten (*compileren*) naar een uitvoerbaar bestand. Dat compileren is tijdrovend en er kunnen gemakkelijk fouten bij optreden.

Verschillende personen kwamen tot de conclusie dat steeds zelf compileren niet ideaal was. Ze besloten *distributies* te gaan uitgeven, dat is een pakket met een kernel, utility's, shells, en vele programma's, allemaal 'voorgecompileerd' dus direct gebruiksklaar. In die begintijd waren bijvoorbeeld de *Slackware Linux* en *SLS Linux* distributies goed bekend. Meer recente distributies zijn *Caldera OpenLinux*, *Red Hat Linux*, en de daarvan afgeleide *Linux Mandrake* en *SuSE Linux*. Hoewel tijdens het gebruik weinig verschillen merkbaar zijn (het is tenslotte allemaal GNU/Linux), lopen installatie- en configuratiemogelijkheden sterk uiteen.

2.4 Debian GNU/Linux

Debian GNU/Linux is een relatief nieuwe distributie, die in 1993 is gestart door Ian Murdock. De belangrijkste reden om met Debian te beginnen was ontevredenheid met alle bestaande distributies. Die werden namelijk door één of enkele personen (een gesloten groep) samengesteld. Allerlei fouten (*bugs*) en onnauwkeurigheden traden op, simpelweg omdat de distributeurs (*distribution maintainers*) geen tijd hadden om alle fouten op te sporen en te verhelpen. Bovendien waren er vaak problemen met het installeren, weghalen en vervangen van software, omdat een echt goed systeem hiervoor ontbrak.

Vanaf de start is Debian ontwikkeld zoals de Linux kernel: iedereen mag meehelpen, het is geen 'gesloten groepje'. Er zijn momenteel ruim 450 *developers*, meer dan bij welke andere Linux distributie ook.

Voor Debian is een compleet nieuw en zeer uitgebreid *packaging system* (pakket-systeem) ontwikkeld. Ieder programma of groepje programma's zit in een uniform *package* of *pakketje*. In elk package is ook vastgelegd hoe het geïnstalleerd moet worden, hoe het weggehaald moet worden, welke andere pakketjes het nodig heeft, en welke pakketjes er totaal niet mee kunnen samenwerken. Debian 2.2 heeft meer dan 4400 pakketjes. De programma's **dpkg** en **apt** kunnen pakketjes moeiteloos installeren en weghalen.


```
// Dorpsnet adres -> naam (let op omgekeerde 192.168.1)
zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.dorpsnet-rev";
};
```

Als je alle bestanden naar je zin veranderd hebt, moet je dat weer aan **named** doorgeven met **/etc/init.d/bind reload**.

Met **nslookup rocky.dorpsnet.nl** en **nslookup 192.168.1.3** kun je nu kijken of heen- en terug-zoeken goed werkt (Ctrl-C als het te lang duurt = fout gaat). Mochten er problemen zijn, dan staat in **/var/log/daemon.log** meestal een foutmelding met een indicatie van de oorzaak.

Je kunt alle vragen aan de nameserver laten loggen in **/var/log/daemon.log**; dat is handig bij problemen, maar kan erg lange logfiles veroorzaken. Daarvoor moet je in **/etc/bind/named.conf** in de sectie 'logging' de regel "category queries { default_syslog; };" toevoegen en weer doorgeven met **/etc/init.d/bind reload**.

Merk op dat je ook meerdere 'zones' met één nameserver kunt beheren (gewoon meer bestanden van het type **db.dorpsnet** en **db.dorpsnet-rev**), maar meestal zal dat niet nodig zijn.

Meer informatie staat in de manpage van **named**, in DNS-HOWTO en in **/usr/share/doc/bind**. Let echter op dat hier en daar nog oude configuratie-opties worden gemeld, die met de nieuwe BIND niet meer werken. Bovendien komen veel bestandsnamen niet overeen met die op een Debian systeem.