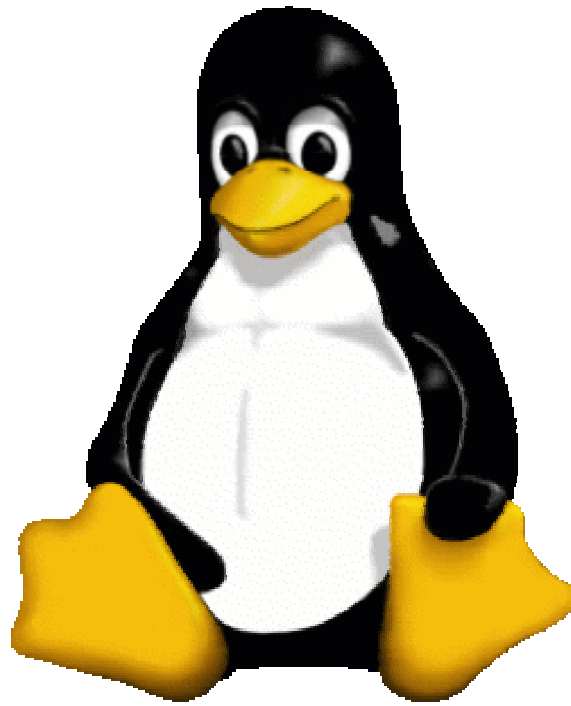


LINUX



Alain Wenmaekers

alain.wenmaekers@skynet.be

Inhoud

Cursus Linux

Inhoud.....	2
Licentie (FDL).....	3
Inleiding.....	4
Structuur.....	10
Bestandssystemen.....	16
Unix commando's.....	22
Oefeningen UNIX (start).....	28
Oefeningen UNIX (bestandssysteem).....	33
Oefeningen UNIX (toegangsrechten).....	41
Oefeningen UNIX (editors).....	42
Software installeren.....	46
Gebruikersbeheer.....	48
Procesbeheer.....	52
Het onderliggend systeem.....	55
IP.....	61
X-Window System.....	72
Scripts.....	77
Internet.....	81
Office.....	92
Servers.....	107
Security.....	120
Bestandsdeling (File Sharing).....	130
NIS.....	141
Virtuele computers.....	144
WINE.....	148
Kernel compile.....	149
Credits.....	152

Licentie (FDL)

Copyright (c) Alain Wenmaekers.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation;

A copy of the license can be found on <http://www.gnu.org/licenses/fdl.html>, "GNU Free Documentation License"

This document is created in *Openoffice 1.0.1*. The course is available in OpenOffice and PDF format.

V1.10

Inleiding

Wat is linux

Linux is een Unix-achtig besturingssysteem. Als we het strikt nemen bestaat Linux alleen maar uit het hart van dit operating system (de kernel). Meestal bedoelen we met Linux de kernel en alle software die reeds geschreven werd voor die kernel.

Duizenden programmeurs werkten gedurende jaren vrijwillig aan het besturingssysteem. Tegenwoordig worden er veel van deze mensen aangeworven bij bedrijven die met Linux bezig zijn. Zowat alle sleutelfiguren die ooit vrijwillig gewerkt hebben aan Linux zijn nu in dienst van deze bedrijven. Dit was belangrijk voor de doorbraak van Linux.

Historiek (Unix)

Het ontstaan van Unix situeert zich in de jaren '70 bij AT&T Labs, waar een systeem werd geschreven voor een mainframe. AT&T voerde een vrijgevig beleid met betrekking tot de licentierechten van de UNIX broncode ten opzichte van universiteiten en onderzoeksinstellingen. Omwille van deze beschikbaarheid is UNIX opgegaan in een reeks van besturingssystemen die niet compatibel waren met elkaar. Men kent wel twee stromingen: BSD (Berkeley Software Distribution) en SysV (AT&T). Fabrikanten zoals Sun, IBM, DEC, SCO, and HP wijzigden hun Unix variant om hun product te onderscheiden van anderen. Dit leidde gedeeltelijk tot een versplintering van Unix, maar niet in die mate als meestal aangenomen wordt.

Schema: zie appendix

Enkele voorbeelden van Unices die gebaseerd zijn op de AT&T code: HP-UX (Hewlett Packard), SunOS/Solaris (Sun), IRIX (Silicon Graphics), Digital Unix/Tru64 (Compaq), SCO (SCO/Caldera), AIX(IBM),...

Ondertussen zijn er verschillende projecten die op de UNIX-filosofie gebaseerd zijn, maar geen AT&T broncode bevatten zoals Linux, *BSD (FreeBSD, OpenBSD, NetBSD) en GNU Hurd.

Cursus Linux

Historiek (Linux)



Linux is ontstaan toen in 1991 een Finse student, Linus Torvalds, een OS (operating system) ontwierp dat gebaseerd was op Minix. Minix was een soort baby-Unix, geschreven door Andy Tanenbaum, hoogleraar aan de Vrije Universiteit van Amsterdam. Het Minix-systeem was bedoeld om studenten met een OS vertrouwd te maken.

Linus zocht een oplossing voor een technisch probleem. De computer van zijn universiteit kon niet meer dan 16 gebruikers tegelijk aan. Wat Linus deed was van grond af aan (maar op basis van zijn kennis van Minix) een nieuw OS schrijven.

Linux bleek de uitkomst voor de computeraars die zelf wilden programmeren. Ze konden, omdat de source code vrij te verkrijgen was, zelf aanpassingen maken.

Ondertussen is Linux aanvaard bij systeembouwers zoals IBM, Dell en Compaq als alternatief OS.

Kernel

De kernel is het hart van een OS. Het is een stuk software dat de hardware beheerd (oa. geheugenbeheer, processcommunicatie, processorbeheer,...).

De linuxkernel is met broncode vrij verkrijgbaar op www.kernel.org.

Distributies

Een distributie wordt gemaakt door een bedrijf of een groep personen die een installatieprogramma, de linuxkernel en bijhorende software op een informatiedrager (CD, DVD, internet) plaatst.

Enkele voorbeelden:

- **RedHat:** de meest gekende distributie. Het is Amerikaans van oorsprong, maar wordt ook vaak in Europa en Azië gebruikt
- **Mandrake:** Frans van oorsprong. Deze distributie is vooral gericht op gebruiksvriendelijkheid
- **SuSE:** een Duitse versie van linux die populair is in Europa
- **TurboLinux:** Aziatische distributie
- **Caldera:** Eigenaar van SCO, en maker van oa. DR-DOS
- **Debian:** gemaakt door vrijwilligers
- **Slackware:** gelijkaardig met Debian
- **Yellow Dog:** Linux voor de Mac



Cursus Linux

*BSD

Vaak hoort men in linux nieuwsgroepen ook de namen van FreeBSD, OpenBSD en NetBSD vallen. De *BSD's zijn net zoals linux vrij verkrijgbaar. Vaak is het verschil met een linuxsysteem heel klein omdat de *BSD's gebruik maken van dezelfde programma's (de kernel is het enige grote verschil tussen deze twee systemen).

- **FreeBSD:** gemaakt om als snelle server te dienen
- **NetBSD:** gemaakt om op zoveel mogelijk verschillende systeemarchitecturen te draaien
- **OpenBSD:** op een bepaald moment afgesplitst vanuit NetBSD na een "flamewar" tussen de NetBSD ontwikkelaars en Theo De Raadt. Deze laatste heeft dan OpenBSD opgericht, dat vooral gemaakt is met "security" in gedachte.

GNU

De commercialisering van UNIX stootte een zekere Richard Stallman zozeer tegen de borst dat hij besloot actie te ondernemen. Met de typisch Amerikaanse mengeling van naïviteit en dadendrang richtte hij in 1984 de "Free Software Foundation" op.

Hij schreef samen met zijn geestesgenoten software die op praktisch elke Unix kon draaien.

GNU, wat staat voor "Gnu is Not Unix", is de verzamelnaam van deze Unix-software. Toen de GNU programma's vervolledigd waren moest er nog een kernel worden geschreven om een volwaardig UNIX systeem te hebben. De GNU Hurd kernel was hiervoor bedoeld maar deze is nooit afgeraakt. De Linuxkernel kwam net op het juiste moment om deze leegte op te vullen.



GPL / BSD licenties

Linux wordt beschermd door de GNU Public License (GPL). Hoofdzakelijk houdt de GPL in dat de broncode van Linux altijd vrij beschikbaar moet zijn. Iedereen kan aanpassingen maken, maar de broncode van deze aanpassingen moeten ook vrij beschikbaar blijven.

Linux is niet enkel "free" in de zin van "free beer" (gratis), maar ook als "free speech" (vrij).

BSD (Berkeley) is een andere open source licentie. Deze stelt dat iedereen de broncode mag nemen en eventueel gebruiken in commerciële software zonder de aanpassingen terug te geven aan de "gemeenschap". (een voorbeeld hiervan is het gebruik van BSD code voor de TCP/IP implementatie in Windows).

Cursus Linux

Kenmerken van Linux

Stabiliteit

Linux is veel stabielere dan Windows. Ook in Linux kan een programma vastlopen, maar het zal niet je hele besturingssysteem vellen. Er zijn veel mensen die Linux maanden en zelfs jaren draaiende houden op hun PC zonder één enkele reboot.

Prijs

Naast het feit dat je Linux gratis kan downloaden of kopiëren kan je voor weinig geld een distributie aanschaffen. Je krijgt dan één of meerdere cd's boordevol software.

Opmerking: uiteraard hebben veel mensen Windows illegaal gekopieerd maar in het geval van Linux is dit 100% legaal.

Open Source

Hier draait alles om bij Linux. Ben je een programmeur en wil je de software aan je noden aanpassen, geen probleem. Je hebt toegang tot de broncode. Nadien moet je natuurlijk op jouw beurt de code ter beschikking stellen voor anderen. Door het systeem van *open source* kunnen bugs heel snel opgespoord worden, tenslotte mag iedereen aan de software werken.

Multi-Tasking Multi-User

Linux is een multi-tasking en multi-user besturingssysteem. Terwijl je een groot document afprint, een cd schrijft en een DVD afspeelt zal Linux niet gaan vertragen. Er kunnen ook meerdere gebruikers terzelfdertijd van één computer gebruik maken. De configuratie- en persoonlijke bestanden staan per gebruiker in een eigen directory.

Platform onafhankelijk

Linux is portable. Dit houdt in dat Linux werkt op PC's, Apple Macintosh, Sun werkstations, Alpha computers,...

Scalable

Het toepassingsgebied is enorm uitgebreid. Linux draait van van handhelds (zie figuur van agenda), mainframes (zie figuur van de IBM z/900) tot op je PC.



Cursus Linux

Wat doe je ermee?

De mogelijkheden van Linux zijn **onbeperkt**. Je kan de hele configuratie van jouw PC volledig zelf instellen. Je verkrijgt de totale controle over jouw pc met de daarop draaiende software.

Je kan Linux gebruiken in de commandline mode zoals de meeste wel kennen uit het *DOS*-tijdperk of in een grafische omgeving zoals in *Windows* (verder dan dat gaat de vergelijking tussen beide systemen niet op).

Je kan Linux gebruiken als web-, proxy-, mail- of DNS-server. Of gewoon als kleine server voor je LAN-netwerkje thuis met daarop een print- en bestandsdeling.

Wie kan linux gebruiken?

Grote bedrijven

Bij de grote bedrijven is Linux reeds ingeburgerd. Dit is normaal, daar er sprake is van grote netwerken (met bijhorende servers) en het ook daar is dat Linux zich opperbest voelt. Vaak maken deze bedrijven eveneens gebruik van Unix.

KMO's

Ook kleine en middelgrote ondernemingen zijn gebaat bij het gebruik van Linux. Zij kunnen het zich dikwijls niet permitteren software te draaien zonder de nodige licenties. Dit betekent vaak een grote kost. Het brengt eveneens met zich mee dat de software niet regelmatig upgedate wordt. Voor hen kan Linux een grote besparing betekenen.

Thuisgebruikers

Daar de thuisgebruiker zelf een echte all-roundgebruiker is van de PC moet iedereen voor zichzelf uitmaken of Linux voor hem/haar nuttig is. Voor sommige zaken kan je niets doen met Windows maar het omgekeerde is ook waar.



Fun & informatief

www.slashdot.com (discussiegroep)

www.linux.com (linux info)

www.linuxdoc.org (handleidingen)

www.freshmeat.net (software)

www.userfriendly.org (cartoon)

www.linuxbelgium.net (startpagina)

www.deja.com (nieuwsgroepen)

www.linux.org (linux info)

www.mandrake.com

www.redhat.com

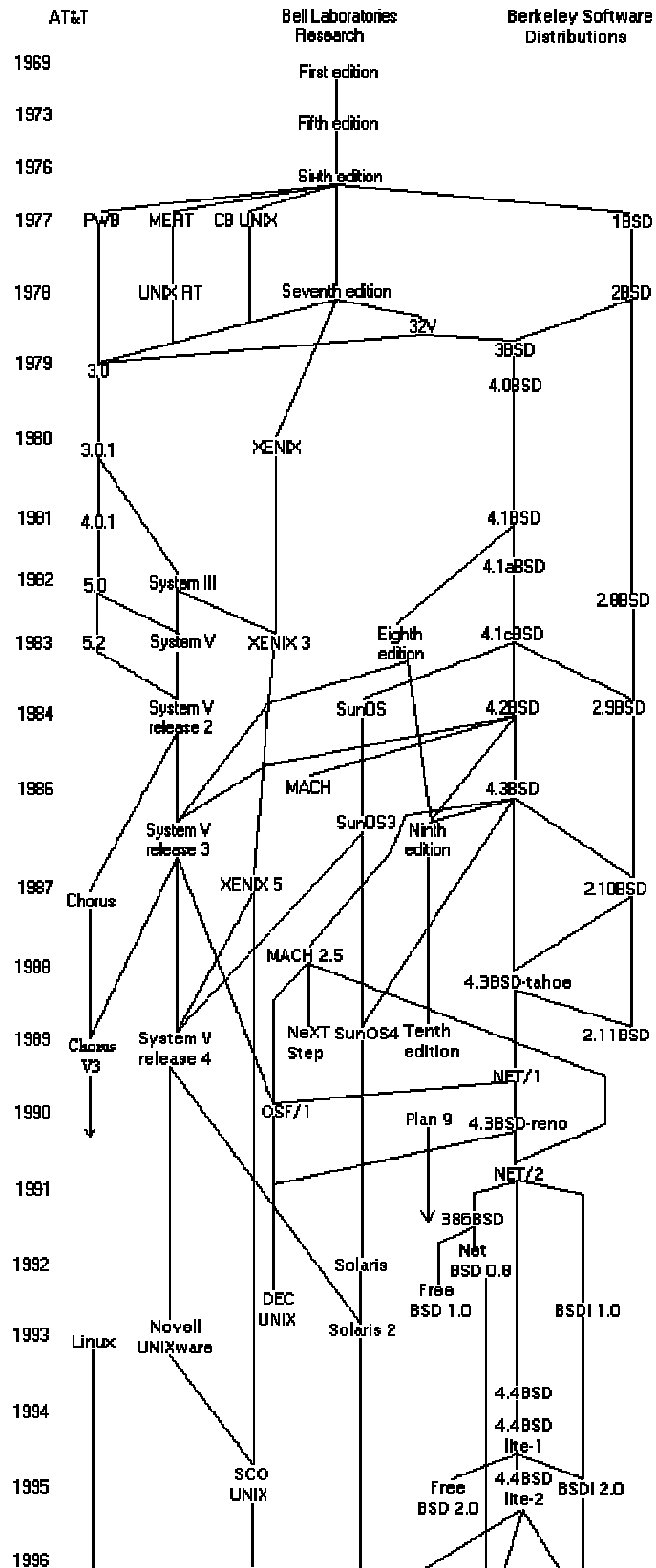
www.ibm.com/linux

www.linuxdevices.com

Cursus Linux

Appendix

De geschiedenis van UNIX



Structuur

*Linux, Windows and DOS
(also known as the Good, the Bad and the Ugly)*

Unix filosofie

Unix moet je je voorstellen als een bijzonder rijk gesorteerde gereedschapskist waar alle werktuigen gespecialiseerd zijn in een bepaalde taak of handeling. Door een aantal van deze kleine tools te laten samenwerken, kunnen grote en ingewikkelde bewerkingen worden uitgevoerd. De gebruiker moet zelf voor elke taak een coherent geheel samenvoegen.

Partities

Linux kan op verschillende manieren worden geïnstalleerd. De meest gangbare manier is het gebruik van lege partities. Deze partities formateer je dan met de "installer". Een andere manier is het gebruik van een "loopback systeem". Hiermee kan je het volledige bestandssysteem wegschrijven naar een bestand (dat zich bevindt op bijvoorbeeld een windowspartitie). Het gebruik van een "loopback" systeem heeft nadelige gevolgen voor de systeempersormantie.

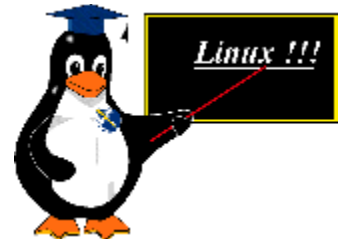
Swap

De swap wordt gebruikt indien het geheugen volloopt. De computer gaat dan data die niet frequent wordt gebruikt wegschrijven naar de vaste schijf. Dit systeem vertraagt de computer, maar kan voorkomen dat het RAM geheugen volloopt.

Linux vereist minstens één swap partitie. Om performantieredenen mag je een aantal swap partities spreiden over meerdere harde schijven. Het is mogelijk om linux te gebruiken met een swapbestand (zoals in Windows), maar geen enkele linuxdistributie ondersteunt standaard deze "setup".

Een swap partitie heeft ten opzichte van een swapbestand enkele voor- en nadelen. Een aparte swap partitie is een stuk sneller omdat deze gestructureerd is om als swap te dienen. Daarentegen is een swapbestand flexibel omdat de grootte en plaats kan worden aangepast zonder wijzigingen te maken aan de partities.

Nota: Linux ondersteunt het "at runtime" wijzigen van de partities.



Bestandsstructuur

Net zoals Windows heeft Unix een hiërarchische bestandsstructuur. Het startpunt noemt men de *root* (wortel, voorgesteld door "/"). Hieronder bevinden zich de mappen (of "directories"). In mappen kunnen bestanden of andere mappen worden geplaatst.

In Linux bestaan er geen "Drives" (vb. "c:"), maar er worden andere schijven bevestigd (mounting) binnen het bestandssysteem. Praktisch voorgesteld zou een subdirectory eventueel een andere schijf kunnen zijn.

Nog een verschil met DOS/Windows is dat Windows gebruik maakt van de "\" (backslash) terwijl het bij Linux/Unix een "/" (slash) voorstelt.

Er is een indeling die op zowat ieder Unix systeem terug te vinden is:

- /dev Speciale bestanden voor de benadering van hardware
- /bin Deze directory bevat de meest essentiële hulpprogramma's
- /sbin De meest essentiële hulpprogramma's voor de administrator
- /etc Hier worden de configuratiegegevens bijgehouden
- /tmp Programma's kunnen hier hun tijdelijke bestanden schrijven
- /lib Bibliotheken voor programma's
- /boot Plaats waar de kernel wordt opgestart
- /var Various, meestal worden hier logbestanden ingeplaatst
- /home Directory waar gebruikersdata in terecht komt
- /usr Plaats waar meestal programma's in terecht komen
- /proc Verwerking van systeeminformatie. Enkel beschikbaar op Linux
- /mnt Gangbare plaats waar bestandssystemen worden aangekoppeld



Speciale mappen

Binnen Linux bestaan er twee speciale mappen: */dev* en */proc*. De */dev* map dient om rechtstreeks randapparatuur aan te spreken.

De */proc* map kan worden gebruikt om configuratiegegevens uit te halen (hoeveelheid vrij geheugen, hardwareinfo, status van het systeem,...), maar ook om "at runtime" systeemparameters te wijzigen (routing aan/uitzetten, gebruik van de hardware aanpassen,...).

Virtuele consoles

De gebruiker heeft meerdere consoles ter beschikking. De toetsencombinatie *CTRL-ALT-Fx* ($1 \leq x \leq 6$). Je kan op deze manier meerdere keren inloggen en dus ook meerdere programma's tegelijk draaien.

In grafische mode heb je ook een gelijkaardige werking, maar daar moet je meerdere grafische omgevingen opstarten om er gebruik van te maken. Voor de grafische omgevingen is *CTRL-ALT-Fx* ($7 \leq x \leq 12$) gereserveerd. Men kan met deze toetsencombinaties ook schakelen tussen tekst en grafische mode.

Gebruikers

Voordat je Linux kan gebruiken moet je eerst inloggen als een *user* (gebruiker) met een paswoord. Aan de hand van de gebruikersnaam kan de computer weten welke dingen de gebruiker van het systeem mag doen.

Naast de gebruikersnaam heb je ook *groups* (groepen). Je kan bepaalde rechten toekennen aan een groep, die groep kan je dan aan iedereen toekennen die de rechten op het systeem nodig heeft.

Er is één gebruiker die rechten heeft op alles, deze wordt "*root*" genoemd.

Iedere gebruiker heeft zijn eigen *home-directory*. In deze map komen alle persoonlijke gegevens terecht alsook het gebruikersprofiel van alle applicaties.

Unix bestandspermissies



Windows NT, Moderne Unices en Linux ondersteunen ACL (Access Control Lists). ACL is een nieuwere vorm voor het toekennen van gebruikersrechten op bestanden. Het heeft als voordeel dat je permissies van meerdere gebruikers en groepen kan koppelen aan een bestand (het is flexibel). ACL onder Unix/Linux is een uitbreiding op de Unix bestandspermissies. Jammer genoeg worden de meeste linuxdistributies geleverd zonder ACL ondersteuning. Zelfs indien er ACL voorhanden is, kan het nog altijd nuttig zijn om kennis te hebben van de structuur van Unix toegangsrechten.

Aan een bestand zijn er drie soorten gebruikers: *owner*, *group*, *other*.

- Owner: Staat voor de gebruiker zelf. De gebruiker kan instellen welke rechten hij/zij op de bestanden heeft.
- Group: Je kan hiermee instellen welke rechten de gebruikers hebben die in dezelfde groep zitten als jij.
- Other: Hier kan men de toegangsrechten instellen van iedere andere gebruiker.

Aan ieder van deze drie soorten gebruikers kan je volgende rechten toekennen: *read* (lees), *write* (schrijf) en *execute* (uitvoeren).

Daemons

Een *daemon* is een programma dat op de achtergrond bewerkingen uitvoert. Het zijn processen die worden gestart bij het booten en ze maken handelingen die belangrijk zijn voor de functionaliteit van het systeem. Normaal gezien hebben deze geen interactie met de gebruiker. Enkele voorbeelden van daemons: *inetd* (netwerkserver), *crond* (het regelmatig uitvoeren van taken), *syslogd* (systeemlogging), *gpm* (muis in de console), *httpd* (webserver),...



Systeeminitialisatie

Iedere keer een linuxsysteem start worden er een aantal initialisatiescripts gestart. Men kan deze scripts vinden in */etc/init.d*. De scripts worden gebruikt om *daemons* op te starten. Er is ook een soort "batch-bestand" waar je zelf een aantal dingen kan in plaatsen die moeten gestart worden (in RedHat en aanverwante distributies is dit */etc/rc.d/rc.local*). Opstartscripts kunnen enkel gewijzigd worden door "root".

Naast de systeemscripts bestaan er ook bestanden die aangeven wat moet gedraaid worden indien een gebruiker aanlogt. In */etc/bashrc* kan men bewerkingen zetten die voor ieder gebruiker geldig zijn, *\$HOME/.bashrc* (*\$HOME* = de homedirectory) dient om bewerkingen uit te voeren voor die bepaalde gebruiker die eigenaar is van die homedirectory. Iedere gebruiker kan voor zichzelf dit bestand aanpassen.

Bestandsnamen en speciale tekens

Unix laat toe om bestandsnamen te gebruiken met zowat alle karakters, maar algemeen aangenomen vermijd je spaties, tabs en karakters met een speciale betekenis zoals:

& ; () | ? \ ' " ' [] { } < > \$ - ! /

De lengte van een bestandsnaam hangt af van het gebruikte bestandssysteem. Zowat ieder systeem ondersteunt een bestandsnaam van 256 karakters of meer.

Het is ook van belang te weten dat bestandsnamen **hoofdlettergevoelig** zijn, bijvoorbeeld *file* en *FILE* wordt aanzien als een verschillend bestand.

Bestanden die beginnen met een "." worden als **verborgen** beschouwd.

Bij het opvragen van bestanden zijn er nog enkele "wildcards" die kunnen gebruikt worden.

- * Komt overeen met een willekeurig aantal opeenvolgende tekens
- ? Eén willekeurig teken
- [...] Specificeert een teken uit de door vierkante haken omsloten verzameling. Bijvoorbeeld `[a-z]` wil een teken uit de verzameling letter van a tot z zeggen.

Bijkomende lijst met speciale tekens en hun betekenis:

- \ Het teken dat na de "\" komt wordt letterlijk geïnterpreteerd
- ' Aanhalen van een string
- " Aanhalen van een string en commando substitutie
- ` (backquote) Commandosubstitutie
- \$ Referentie naar een (shell)variabele
- {} Commandogroepering in een functie (bij scripts)
- # Commentaar
- & Voert een commando uit in de achtergrond
- () Groepeert commando's of identificeert een functie
- | Pipe
- > Redirection
- >> Redirection als toevoeging
- < Redirect vanuit de standaard input

Redirection & Pipes

Unix laat toe om de uitvoer van het ene commando als input te gebruiken voor het andere. Dit noemt men een "pipe".

Bijvoorbeeld: `ls | more` (opvragen inhoud map, met gebruik van pagina's).

Pipes laten toe om meerdere programma's aan mekaar te koppelen (denk aan de gereedschapskist waarmee je complexere commando's kan opbouwen)

Met redirections kunnen we de uitvoer van een programma naar een bestand schrijven.

Bijvoorbeeld: `ls > lijst` (we schrijven de lijst van de dir in het bestand "lijst").

Er bestaat ook een ">>" teken dat bijna dezelfde functie heeft als ">". Bij ">" wordt eerst het bestand leeggemaakt, met ">>" wordt er achteraan het bestand verdergeschreven.

Environment variabele

Je kan in de shell een aantal variabelen definiëren waar de shell of externe programma's rekening kunnen mee houden.

Bijvoorbeeld: `export PATH=/bin`

Je kan deze variabele gebruiken in zelf geschreven programma's of scripts. Je kan ze oproepen via *\$naam*.

Bijvoorbeeld: `echo $PATH`



Achtergrondprocessen

Als je een commando start met een "&" teken achteraan, dan wordt dit uitgevoerd op de achtergrond. De console is dan vrij om andere commando's in te geven terwijl dat programma draait. Als men uitlogt, blijft het programma nog verder draaien.

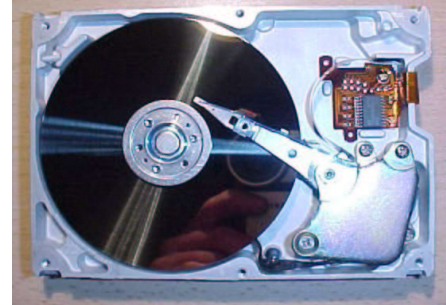
Bestandssystemen

Net zoals elk ander besturingssysteem moet voor Linux een bestandsstructuur aangebracht worden op de harde schijf.

Je hebt bij Linux minimum 2 partities nodig:

- één voor het bestandssysteem zelf (root).
- één voor het virtueel geheugen (swap).

Meestal worden er meerdere partities aangemaakt, om het systeem nog beter te beschermen tegen vastlopen of dataverlies.



Schijven

De bestandsstructuur van Unix/Linux is gemaakt dat randapparaten gekoppeld (gemount) worden aan de bestandsstructuur. Een randapparaat kan dan ook perfect aangesproken worden alsof het een bestand is. Op de achtergrond (eigenlijk in de kernel; ingebakken of als module) draait dan de driver voor dat randapparaat.

Al deze speciale bestanden (randapparaten) bevinden zich in een speciale map **/dev** (devices). Harde schijven moeten aangesproken worden als **/dev/hd x** waarbij **x** een letter [a-z] is, die overeenkomt met de plaats waar de harde schijf wordt aangesloten.

Voor normale IDE-systemen, met twee IDE controllers, waaraan 2 toestellen kunnen aangesloten worden, kun je maximaal 4 harde schijven aansluiten:

IDE1 (primary)	Master	/dev/hda
	Slave	/dev/hdb
IDE2 (secondary)	Master	/dev/hdc
	Slave	/dev/hdd

Ook als er een CDROM of een ander medium is geïnstalleerd, krijgt deze hetzelfde speciale bestand toegewezen in **/dev**.

Cursus Linux

Beschik je over een SCSI-systeem, dan kan je deze harde schijven aankoppelen:

/dev/sda	of	/dev/sr0
/dev/sdb		/dev/sr1
...		... (tot 15)

SCSI CDROM's krijgen de *devices* `/dev/scd x` toegewezen (waarbij x een getal is tussen 0 en 15). SCSI tapes worden aangeduid met `/dev/st x` .

Opgelet: Bij een CD-schrijver wordt de SCSI-emulatie gebruikt om CD's te kunnen schrijven (deze wordt automatisch aangezet bij recente distributies). In dit geval moet je het toestel gebruiken alsof het een SCSI-apparaat is (vb. `/dev/scd0`)

Partities

Een partitie is eigenlijk een stuk van een harde schijf waar je het bestandssysteem op plaatst.

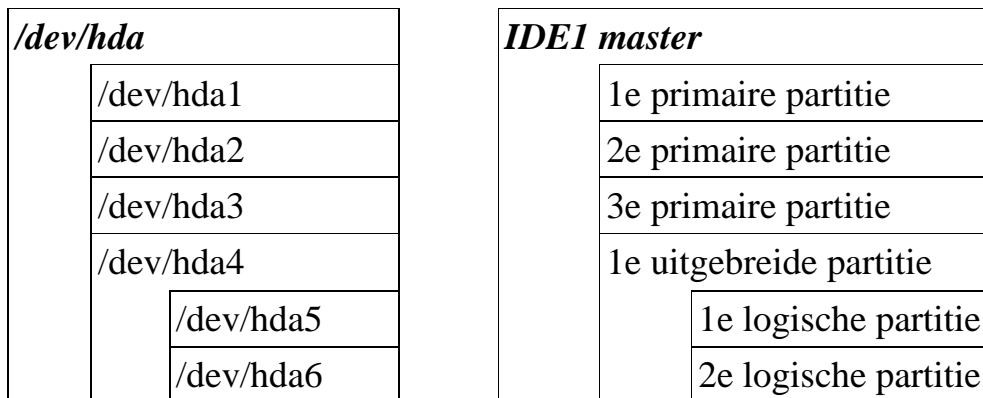
Op een X86 systeem bestaan er 3 soorten partities:

- primaire partitie
- logische partitie (logic)
- uitgebreide partitie (extended)

Een aantal tools om partities te wijzigen: *fdisk*, *cfdisk*, *sfdisk*,...

Er kunnen maar **vier** *primaire partities* zijn. In sommige gevallen is dit weinig. Men heeft daarop een eenvoudige oplossing voor gevonden. De vierde partitie wordt een uitgebreide *partitie*. Daarin kunnen meerdere *logische partities* geplaatst worden.

In Linux kan je deze structuur voorstellen zoals in onderstaande tekening:



Bestandssystemen

Er bestaan een hele hoop bestandssystemen die voor linux geschreven zijn. Vooral het laatste jaar zijn er enkele nieuwkomers die een geweldige "feature list" kunnen tonen.

Er zijn drie soorten bestandssystemen.



Klassieke bestandssystemen

Allereerst zijn er de **klassieke** bestandssystemen, die gebaseerd zijn op *inodes* of een *file allocation table*. Dit zijn delen op de schijf die aangeven waar op de schijf bestanden zijn weggeschreven. Het wegschrijven gebeurt zonder enige gegevensbescherming.

Voordelen:

- snel
- simpel

Nadelen:

- fragmentatie (verdeling van grote bestanden over de gehele schijf)
- geen gegevensbescherming (bij beschadiging van informatie kan er geen recuperatie van de gegevens meer gebeuren).

Voorbeelden zijn FAT (Windows), minix en ext2 (Linux).

Modernere versies hebben wel een verbetering ten opzichte van fragmentatie en gegevensbescherming.

Minix: Het allereerste bestandstelsysteem dat beschikbaar was voor Linux omdat het in de beginjaren afhankelijk was van een Minix installatie. Het minix-bestandstelsysteem komt uit Minix, geschreven door Andy Tanenbaum. De structuur is verouderd en het is afgeraden dit systeem nog te gebruiken.

FAT: Dit systeem wordt gebruikt in DOS en Windows9x/ME. Er zijn meerdere verbeteringen binnen dit systeem die compatibel zijn met elkaar (onder andere FAT-16 en FAT-32, deze laatste was een verbetering om grotere schijven te kunnen adresseren).

FAT heeft geen bescherming tegen fragmentatie of gegevensverlies. Het heeft ook geen implementatie in verband met security.

Linux heeft een zeer goede ondersteuning voor het lezen en schrijven op een FAT-partitie.

Ext2: Ext2 is al lange tijd hét bestandstelsysteem van Linux. Het heeft een beveiliging tegen datacorruptie door meerdere keren de *superblock* weg te schrijven (superblok = data die verwijst naar de inhoud van de schijf). Het probeert ook fragmentatie tegen te gaan (er is geen defragmentatietool beschikbaar).

Softupdates

Het *softupdates* systeem wordt vooral gebruikt in het aangepaste UFS systeem in *BSD. *Softupdates* is een simpele manier om gegevensbeveiliging te bekomen.

Voor er data weggeschreven wordt, gaat het systeem dit melden in een logbestand. Als de data effectief weggeschreven is wordt het logbestand opnieuw aangepast.

Dit komt in ruwe lijnen overeen met *journaling* (het verschil zit vooral in de technische details). Aanhangers van dit type bestandssysteem beweren dat het sneller werkt dan *journaling*. UFS wordt ondersteund in Linux, maar er is geen "native" implementatie in linux (er is wel een bestandssysteem in de maak dat TUX2 heet).

Journalled file systems

Journalled bestandssystemen zitten iets ingewikkelder in mekaar dan *softupdates*. De gegevensbeveiliging gaat ook veel verder, en in de meeste gevallen is de snelheid beter dan een klassiek/*softupdates*. Er zijn voor linux niet minder dan 4 journalled file systems.

Indien er een crash is van een journalled file system, moet er bij de reconstructie van de gegevens geen *chkdsk/fchk* meer uitgevoerd worden.



NTFS: Het bestandssysteem van Windows NT/2000/XP. Dit bestandssysteem wordt niet goed ondersteund door Linux. De driver ondersteunt zonder problemen het lezen van de partities, maar het is nog niet veilig om op NTFS te schrijven.

Ext3: De gelijkenissen met ext2 zijn groot. Ext3 is eigenlijk een ext2 partitie met een bijkomende *journal*. Ext3 partities kunnen zonder problemen gelezen worden door systemen die enkel ext2 ondersteunen. Andersom is het ook mogelijk om ext2 partities om te zetten naar ext3 door het bijvoegen van een journal log.

Reiser: Reiser was het eerste journalled file system voor Linux. Dit systeem heeft zich sinds een lange tijd bewezen heel stabiel te zijn. Er zijn in Reiser enkele nieuwigheden geïmplementeerd die voorheen ondenkbaar waren. Reiser is enorm performant op bewerkingen met een groot aantal kleine bestanden. Er is ook een "tail" optie om meerdere bestanden in één inode te plaatsen wat diskruimte spaart.

- JFS:** Origineel was dit het bestandssysteem van AIX. IBM heeft besloten dit systeem te herschrijven voor Linux. JFS heeft zich al bewezen als één van de beste bestandssystemen in de Unix wereld.
- XFS:** Gelijkaardig met JFS heeft SGI (Silicon Graphics) zijn IRIX bestandssysteem herschreven om op Linux te gebruiken. XFS heeft een uitzonderlijke reeks van "features". Eén van de kenmerken is dat het geoptimaliseerd is om met grote bestanden te werken.

Mounting

Een bestandssysteem aankoppelen noemt men "mounten". Het vreemde bestandssysteem wordt dan opgenomen binnen de Unix-boomstructuur. Meestal wordt dit gedaan in de `/mnt` map, maar als de gebruiker dit wenst kan het ook een andere (lege) map zijn.

Om een CD-ROM te mounten gebruiken we (in volledige notatie):

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

De optie "-t iso9660" specificeert dat we een bestandssysteem willen aankoppelen dat conform is met de cd-standaarden. De inhoud van de CD zal te vinden zijn in `/mnt/cdrom`. Zonder de optie "-t" zal mount het bestandssysteem zelf proberen te detecteren. Twee andere voorbeelden:

- Vaste schijf: `mount /dev/hda1 /home`
- Floppy: `mount -t vfat /dev/fd0 /mnt/floppy`



Indien men een wisselbare schijf aangekoppeld heeft moet men deze eerst "unmounten" voordat ze kan verwijderd worden.

Nota: Vele fabrikanten van distributies hebben enkele gebruiksvriendelijke aanpassingen gemaakt. Zo kan men meestal een CD-ROM of floppy laden door respectievelijk "mount `/mnt/cdrom`" en "mount `/mnt/floppy`".

In de grafische omgevingen van Linux zijn er ook gemakkelijkere manieren om te mounten, in de meeste gevallen gebeurt dit zelfs automatisch.

fstab

In dit bestand (`/etc/fstab`) staat een lijst welke bestandssystemen gekend zijn voor het systeem en welke automatisch worden aangekoppeld.

Een voorbeeld:

LABEL=/	/	ext2	defaults	1	1
/dev/fd0	/mnt/floppy	auto	noauto,owner	0	0
none	/proc	proc	defaults	0	0
none	/dev/pts	devpts	gid=5,mode=620	0	0
/dev/hda5	swap	swap	defaults	0	0
/dev/cdrom	/mnt/cdrom	iso9660	noauto,owner,ro	0	0
1	2	3	4	5	6

1 = *devicename* waar het bestandssysteem opstaat

2 = Mountpoint. Locatie waar het bestandssysteem op wordt gekoppeld

3 = Type

4 = Opties (zie "**man mount**")

5 = Dump. Geeft de mogelijkheid om een "dump" aan te maken van het bestandssysteem

6 = Volgorde van "fsck". Controle van de schijven in deze volgorde (0=niet controleren)

De bestandssystemen waarbij als *devicename* "none" staan zijn speciale bestandssystemen. Ze worden door het systeem gegenereerd en worden niet opgeslagen op de schijf.

Bootloader

Na de installatie van Linux zal de computer bij het opstarten een scherm tonen met een keuzemenu waaruit je alle besturingssystemen die geïnstalleerd zijn op de PC kan laden. Als je bijvoorbeeld al Windows hebt geïnstalleerd, dan kan je kiezen tussen *Linux* en *Windows*.

Vanuit dit menu kan je ook verschillende versies van Linux starten (zie hoofdstuk *Kernel Compile*).

De twee meest gekende bootloaders (op het X86 platform) die bij Linuxdistributies geleverd worden zijn **LILLO** en **GRUB**. Beide programma's kunnen zowel via configuratiebestanden als een GUI (vb. Linuxconf) worden geconfigureerd.

Deze programma's hebben tijdens het opstarten geen toegang tot de configuratiebestanden, dus is het belangrijk om na een wijziging een "update" uit te voeren van de configuratie

vb: Bij LILLO vindt je het configuratiebestand in `"/etc/lilo.conf"`. Na een wijziging moet je het commando `"/sbin/lilo"` uitvoeren om de veranderingen effectief te maken.

Unix commando's

Dit hoofdstuk heeft als doel de meest gebruikte Unix/Linux commando's te beschrijven. Na de theorie volgen oefeningen.

cat

De inhoud van een bestand tonen.

Vb: `cat filename`

man

Het opvragen van de handleiding.

Vb: `man cat` (Geeft een handleiding van cat)

cd

Veranderen van de actuele directory.

Vb: `cd /usr/bin`

mkdir

Het aanmaken van een nieuwe directory

Vb: `mkdir map`

ls

Geeft een overzicht van bestanden. Bij gebruik van de optie `-l` krijg je gedetailleerde voorstelling.

cp

Kopieert een bestand of directory

Vb: `cp bronbestand doelbestand`

Je kan ook hele mappen dupliceren door de optie `-r`.

Vb: `cp -r brondirectory doeldirectory`

mv

Verplaatst een bestand of directory. Dit commando kan ook gebruikt worden om bestanden te hernoemen.



chmod

Wijzigen van bestandspermissies

chown

Wijzigt de eigenaar (gebruikersnaam en/of groep) van een bestand.

dd

Converteert en kopieert bestanden. Meestal wordt dit commando gebruikt om "images" te maken van een schijf.

Vb: `dd if=/dev/cdrom of=bestandsnaam.iso`

file

Bepaalt tot wat voor type een bestand behoort.

Vb: `file testbestand.jpg`

find

Zoekt naar bestanden in directorybomen.

Vb: `find . -name testbestand`

locate

Heeft ongeveer dezelfde functie als "find", maar in plaats van de directoryboom te doorlopen werkt locate met een database die regelmatig wordt ge-update.

ln

Maakt een bestand bekend onder een andere plaats/locatie.

Vb: `ln -s bestandsnaam symlinknaam`

De optie "-s" dient om een "soft symlink" te maken. Hierbij is het zichtbaar dat het bestand eigenlijk een link is.

nl

Laat ingelezen regels voorafgaan door regelnummers

pwd

Laat de actuele directory zien

rm

Verwijdert bestanden en/of directorybomen.

Vb: `rm teverwijderenbestand`

Met de optie "-r" kan je ook de mappen en hun inhoud verwijderen.

head

Drukt de eerste regels van een bestand af.

tail

Drukt de laatste regels van een bestand af. Met de optie "-100" (waarbij 100 ook een ander getal kan zijn), druk je de laatste 100 regels af.

De optie "-f" volgt continue het einde van het bestand. Zo kan je op een gemakkelijke wijze bijvoorbeeld logbestanden volgen.

Vb: `tail -f /var/log/messages`

tar

Dit is een tool om backups van bestanden/mappen te maken.

Om bijvoorbeeld een backup te maken van een logdirectory gebruik je:

```
tar -cf log.tar /var/log
```

Om een *.tar* bestand uit te pakken gebruik je:

```
tar -xf log.tar
```

Met de optie -v (verbose) kan je volgen welke bestanden worden gelezen of uitgepakt.

gzip & gunzip

Deze tools dienen om bestanden te compresseren. *Gzip* gebruik je om gezipte bestanden te maken, *gunzip* om ze uit te pakken. *Gzip* kan maar één enkel bestand per keer compresseren, daarom wordt het vaak in combinatie gebruikt met *tar*. Dit verklaart waarom *.tar.gz* bestanden een populair formaat is (tar beschikt ook over de optie -z om automatisch gzip bestanden aan te maken).

touch

Actualiseert de creatie-, benaderings-, en wijzigingsdatum. Het "touch" commando wordt ook vaak gebruikt om een leeg bestand aan te maken.

wc

Word Count. Telt het aantal woorden, tekens en regels in bestanden.

mail

Verzendt en ontvangt elektronische post. *Mail* wordt vooral gebruikt in scripts.

write

Verzendt een boodschap naar een specifieke gebruiker.

Vb: `write root`

Je kan dan teksten typen die bij de andere gebruiker op het scherm komen. Telkens je op ENTER druk wordt die tekst verstuurd. Je sluit het programma af door op CTRL-C te drukken. Indien je geen berichten wenst te ontvangen, gebruik dan het commando *mesg*.

mesg

Geeft/stopt de toestemming dat een andere gebruiker berichten kan sturen naar jouw terminal.

Vb: `mesg n` (geen toestemming) of `mesg y` (wel toestemming)

awk

Patroonherkenningstaal.

cmp

Vergelijkt twee bestanden op gelijkheden.

diff

Zoekt de verschillen tussen twee bestanden

grep

Zoekt naar een bepaalde tekststring in bestanden.

Vb: `grep "string die ik zoek" mijnbestand`

uniq

Verwijdert dubbele regels uit gesorteerde bestanden.

sort

Sorteert de regels van samengevoegde bestanden.

vi

Teksteditor.

passwd

Wijzigen van het paswoord.

echo

Toont een tekststring op het scherm.

lpr

Print de inhoud van een bestand af. Het bestand kan drie formaten hebben: tekst, postscript of raw (taal van de te gebruiken printer).

Vb: `lpr test.ps`

kill

Beëindigt een aangegeven proces.

Vb: `kill 1201` (beëindig het proces met nummer 1201)

Met de optie "-9" kan je aangeven dat het systeem het proces op hardhandige wijze stopt. Gebruik enkel de optie "-9" indien een proces op hol slaat en het niet op een andere manier kan gestopt worden.

killall

Heeft dezelfde functie als *kill*, maar met *killall* wordt een procesnaam gebruikt in plaats van een procesnummer.

ps

Geeft een lijst van de gestarte processen. Met het commando "*ps -ef*" krijg je een lijst van alle processen die op het systeem draaien.

top

Top toont een lijst van de processen die het systeem het zwaarst belasten. Je kan er ook andere statistische gegevens in terug vinden.

df

Het opvragen van een lijst van alle "gemounte" partities. Bij de uitvoer wordt ook getoond hoeveel vrije ruimte er beschikbaar is. Met de optie "-h" krijg je een duidelijkere uitvoer.

du

Toont de hoeveelheid schijfruimte in gebruik voor een opgegeven directoryboom.

nice

Voert een proces uit met een verminderde prioriteit.

date

Toont of wijzig de tijd van het systeem.

su

Dit staat voor "Switch User". Je hebt de mogelijkheid om van gebruiker te veranderen.

Vb: `su gebruiker2`

Indien geen gebruikersnaam wordt opgegeven wordt er vanuit gegaan dat "root" wordt bedoeld. Voor het wijzigen van de gebruiker heb je wel het paswoord van die andere gebruiker nodig.

finger

Opvragen van gebruikersgegevens.

Vb: `finger gebruikersnaam`

mount

Het aankoppelen van externe bestandssystemen. Meestal gebruiken we de verkorte notaties.

Vb: `mount /mnt/floppy` of `mount /mnt/cdrom`

umount

Het afkoppelen van externe bestandssystemen.

Vb: `umount /mnt/floppy` of `umount /mnt/cdrom`

who

Toont wie op het systeem aangelogd is.

clear

Maakt het scherm leeg.

Oefeningen UNIX (start)



Linux commando's leren gebruiken

Deze opgaven moeten je vertrouwd maken met de basisbegrippen van het geven van commando's.

1. Zorg ervoor dat het LINUX venster is geactiveerd.
2. Probeer enkele eenvoudige commando's die geen argumenten of opties vereisen.

<code>date</code>	<i>Toon datum en tijd</i>
<code>pwd</code>	<i>Toon de actuele map</i>
<code>ls</code>	<i>Toon de lijst van bestanden in de huidige map</i>
<code>clear</code>	<i>Maak het scherm leeg</i>
<code>who</code>	<i>Geef een lijst wie aangelogd is op het systeem</i>

3. Probeer nu enkele commando's die argumenten en/of opties vereisen.

<code>ls -a</code>	<i>Toon alle bestanden in een map</i>
<code>ls -al</code>	<i>Toon een lange lijst van alle bestanden in een map</i>
<code>cat .bashrc</code>	<i>De inhoud van .bashrc wordt getoond</i>
<code>mkdir dir1</code>	<i>Maak een map</i>
<code>cd dir1</code>	<i>Wijzig huidige map</i>
<code>cd ..</code>	<i>Ga een map terug</i>
<code>rm -r dir1</code>	<i>Verwijder de map</i>
<code>cp .bashrc b1</code>	<i>Kopieer ".bashrc" naar "b1"</i>
<code>wc b1</code>	<i>Tel het aantal lijnen, woorden en tekens in "b1"</i>
<code>wc -l b1</code>	<i>Tel enkel het aantal lijnen in "b1"</i>
<code>rm b1</code>	<i>Verwijder het bestand "b1"</i>

Cursus Linux

4. Probeer verschillende commando's op één regel in te typen.

```
cp .bashrc testfile; cat testfile
```

Kopieer een bestand en druk de inhoud af

```
ls -l testfile; rm testfile; ls -l testfile
```

Geef de details van het bestand, verwijder het, en toon de details opnieuw



Speciale tekens

Deze oefeningen zullen je vertrouwd maken met het gebruik van verschillende speciale tekens in Linux.

1. Zorg ervoor dat je in je home-directory zit.

Je kan gemakkelijk naar de home-dir gaan met het commando "cd", zonder argumenten.

2. Gebruik de "*" joker om al de bestanden te tonen in de huidige map.

```
ls *
```

(dit commando is gelijkwaardig aan *ls* zonder argumenten)

3. Gebruik het "?" teken om alle bestanden te tonen van vier karakters lang.

```
ls ????
```

Toon alle bestanden van vier karakters

```
touch test
```

Maak een leeg bestand "test"

```
ls ????
```

Toon alle bestanden van vier karakters

4. Concateneer (= voeg samen) drie bestanden tot één bestand. Toon deze dan op het scherm.

echo alpha > a ; echo beta > b ; echo gamma > c

Maak 3 bestanden aan

cat a b c > newfile ; cat newfile

Concatenatie en het tonen op scherm in één commando



Controletoeetsen voor de terminal

Deze oefening zal je vertrouwd maken met de verschillende terminal controletoeetsen die Linux gebruikt.

1. **De commandoregel wissen.** Typ een reeks willekeurige tekens aan de prompt. Druk *niet* op RETURN. Wis de volledige regel door op CTRL-U te typen.
2. **Een opdracht afbreken.** Geef het commando "**sleep 300**" in (doe niets gedurende 300 seconden. Nadat het gestart is kan je eens proberen een ander commando in te geven. Stop nu het *sleep* proces door op CTRL-C te typen. Kan je nu terug commando's intypen?
3. **Tekens wissen.** Typ een reeks van willekeurige tekens aan de prompt. Druk *niet* op RETURN. Probeer nu CTRL-h te gebruiken om tekens te wissen. Vele toetsenborden laten ook toe om "DEL" of "BACKSPACE" te gebruiken.
4. **Autocomplete.** Je kan namen van bestanden automatisch laten vervolledigen. Typ bijvoorbeeld het commando "**cat new**". Druk nu op TAB. Je zal merken dat Linux de bestandsnaam vervolledigd naar **newfile**.



Wijzigen van het wachtwoord

Wijzig je oorspronkelijk wachtwoord. Maak gebruik van het **passwd** programma. Antwoord, zoals gevraagd, met je oude wachtwoord en dan je nieuwe. Bevestig het nieuwe paswoord door het nogmaals in te typen. Merk op dat er op het scherm niets verschijnt wat je typt.



Aan informatie geraken

1. Gebruik het **man** commando om meer te weten te komen over een aantal Linux/Unix opdrachten.

```
man ls
man cp
man rm
man man
```

2. Gebruik de volgende opdrachten om iets te weten te komen over de gebruikers van het systeem.

```
who
who am i
whoami
finger
finger jouw-gebruikersnaam
```



Afmelden van het systeem

Met volgende opdrachten kan je afmelden van het systeem

```
logout  
exit
```

Indien je het systeem wil uitschakelen gebruik je één van volgende opdrachten

```
poweroff      =  shutdown -h now  
reboot        =  shutdown -r now
```



Oefeningen UNIX (bestandssysteem)

ls

Lees de **man** pagina voor de **ls** opdracht

man ls

Gebruik de **ls** opdracht zonder argumenten om de inhoud van folders te tonen.

Hoeveel bestanden worden er getoond?

Probeer ook eens de **dir** opdracht



Gebruik nu **ls** met de **-a** optie. Hoeveel bestanden zie je nu? Merk op dat de bijgekomen bestandsnamen allemaal beginnen met een punt, het zijn dus **verborgen bestanden**.

Recentere linuxdistributies gebruiken kleuren om het onderscheid tussen mappen en bestanden te tonen. Indien je niet beschikt over een kleurenterminal kan de **-F** optie nuttig zijn. Merk op hoe de uitvoer verschilt van de uitvoer van de opdracht zonder argumenten.

Gebruik de vorm `ls -l` om een "lange" lijst van je bestanden te verkrijgen. Een voorbeeld van uitvoer met een verklaring van de getoonde informatie vind je hieronder.

```
-rw-rw-r-- 1 gebruike gebruike      6 jan 11 15:33 a
-rw-rw-r-- 1 gebruike gebruike      5 jan 11 15:33 b
drwxr-xr-x 2 gebruike gebruike 4096 jan 11 14:18 Desktop
-rw-rw-r-- 1 gebruike gebruike      6 jan 11 15:34 g
-rw-rw-r-- 1 gebruike gebruike     17 jan 11 15:34 newfile
-rw-rw-r-- 1 gebruike gebruike      0 jan 11 15:26 test
  1         2         3         4         5         6         7
```

1 = toegangsmodi/permissions

2 = aantal koppelingen

3 = eigenaar

4 = groep

5 = grootte (in bytes)

6 = datum/tijdstip van de laatste wijziging

7 = naam van het bestand

Recursieve lijsten kunnen zeer nuttig zijn. Probeer volgende opdrachten. Wat merk je in de uitvoer?

```
ls -R /usr
ls -Rl /usr
```



more

Lees de **man** pagina voor **more**

Gebruik de **more** opdracht om een bestand te lezen:

```
more /etc/termcap
```

Merk bij het lezen de "**More**" prompt op onderaan de bladzijde.

Probeer op de RETURN/ENTER toets te drukken. Wat gebeurt er?

Druk éénmaal op de SPATIEBALK. Wat gebeurt er?

Typ een letter **b**. Wat gebeurt er?

Gebruik de *voorwaarts zoeken* mogelijkheid om het woord **clockwise** te vinden met de opdracht:

```
/clockwise
```

Ga na welke mogelijkheden er nog zijn door hulp te vragen met het **?**-teken

More gaat verder tot het einde van het bestand of tot je **q** typt om het programma te verlaten. Typ een **q** om het programma te verlaten.



head & tail

Lees de **man** pagina voor de **head** opdracht.

Toon het begin van een bestand met de volgende opdracht. Hoeveel regels tel je?

```
head /etc/termcap
```

Probeer nu deze opdracht en tel het aantal regels dat nu wordt getoond.

```
head -5 /etc/termcap
```

Lees nu de **man** pagina voor **tail**.

Toon nu hetzelfde bestand met de **tail** opdracht. In hoeverre is hetgeen getoond wordt verschillend? Hoeveel regels tel je.

```
tail /etc/termcap  
tail -5 /etc/termcap
```

Ga na waarom de **-f** optie in tail nuttig kan zijn. Vraag het eventueel aan de leraar indien je het niet vindt.



cat

Lees de **man** pagina voor **cat**

Gebruik deze opdracht om de inhoud van een bestand te tonen. Wat gebeurt er?

```
cat /etc/termcap
```

Probeer nu volgende opdracht. Let op het verschil. Hoeveel regels bevat het bestand?

```
cat -n /etc/termcap
```

De **cat** opdracht wordt vaker voor andere doeleinden gebruikt dan het zuiver tonen van een bestand. Gebruik deze opdracht om twee bestanden te concateneren (aan elkaar te koppelen) in een derde bestand.

```
echo One > file1
echo Two > file2
cat file1           Toon inhoud file1
cat file2           Toon inhoud file2
cat file1 file2 > newfile  Voeg bestanden samen
cat newfile         Toon inhoud nieuw bestand
```



cp

Lees de **man** pagina voor de **cp** opdracht.

Kopieer een bestand in de configuratie folder naar een ander bestand in jouw thuis folder en lijst de inhoud om na te gaan of de opdracht gelukt is.

```
cp /etc/termcap mijnbestand
ls
```

Gebruik de copy opdracht met de "inquire" optie. Wat gebeurt er?

```
cp -i /etc/termcap mijnbestand
```

Gebruik de recursieve optie om een volledige subfolder te kopiëren naar een nieuwe subfolder. Lijst de inhoud van beide folders om na te gaan of het heeft gewerkt. (Bij het kopiëren kunnen er toegangsfouten optreden. Dit is normaal).

```
cp -R /usr/bin mijndir
```

De copy opdracht aanvaardt jokers. Probeer volgende opdracht. Wat gebeurt er? (tip: doe een **ls *conf** in *mijndir*)

```
cp /etc/*conf mijndir
```

Indien je een bestand kopieert van een andere locatie naar de huidige folder en je wil de bestandsnaam behouden, kun je "." gebruiken om de huidige folder aan te duiden.

```
cp /etc/issue .
```



mv

Lees de **man** pagina voor de **mv** opdracht

De **mv** opdracht kan gebruikt worden om bestanden te hernoemen. Gebruik deze opdracht en lijst dan de bestanden om te tonen dat de opdracht heeft gewerkt.

```
mv issue issue2
```

De **mv** opdracht wordt ook gebruikt om folders te hernoemen. Probeer deze opdrachten en controleer of deze gelukt zijn.

```
mv mijndir mijnmap
```

Gebruik nu de **mv** opdracht om een bestand te verplaatsen

```
mv mijnbestand $HOME/mijnmap
```



rm

Lees de **man** pagina voor de **rm** opdracht.

Gebruik de **rm** opdracht om een bestand te verwijderen. Lijst de folder uit om na te gaan of het commando geslaagd is.

```
rm newfile
```

Verwijder de map "mijnmap". Waarom gebruiken we de optie **-f**?

```
rm -Rf mijnmap
```

Verwijder nu alle bestanden in jouw thuishmap.

```
rm *
```

Voer een "**ls -al**" opdracht uit. Waarom is niet alles weg? Willen we alles wel weg? Welke opdracht moeten we maken om alles weg te vegen?



file

Lees de **man** pagina van **file**

Gebruik de **file** opdracht om het type van volgend bestanden te bepalen:

```
file /bin/su
file /etc/xinetd.conf
file /bin/*
```



find

Lees de **man** pagina van **find**

Gebruik de opdracht **find** om het bestand *xinetd.conf* te vinden. Doe dit in de */etc* map.

```
cd /etc
find . -name xinetd.conf
```

Zoek nu naar bestanden waarvan de naam eindigt op "conf"

```
find . -name "*conf"
```

Zoek enkel mappen met beginnen met de letters "rc"

```
find . -name "rc*" -type d
```

Leg uit waarom **find** nuttig kan zijn in plaats van **ls**?



ln

Lees de **man** pagina voor de **ln** opdracht

Creëer een koppeling tussen 2 bestanden. Typ daarna de **ls -l** opdracht om te controleren of de opdracht gelukt is. (Vergeet niet naar de home-dir terug te keren)

```
ln -s /etc/issue new.issue
```

Gebruik de **rm** opdracht om de koppeling te verwijderen. Merk op dat de originele bestanden hierdoor niet werden beïnvloed.

```
rm new.issue
```



sort

Lees de **man** pagina van **sort**.

Gebruik de **cat** opdracht om te kijken naar een ongesorteerde namenlijst. Merk op dat er verschillende kolommen zijn en dat de lijst niet alfabetisch geordend is.

```
cp /etc/passwd my.passwd  
cat my.passwd
```

Gebruik nu de **sort** opdracht om een elementaire sortering van het bestand uit te voeren.

```
sort my.passwd
```

Voer dezelfde sortering uit, maar stuur de uitvoer naar een bestand. Gebruik daarna de **cat** opdracht om het bestand te bekijken.

```
sort my.passwd > sorted.passwd ; cat sorted.passwd
```



mkdir

Lees de **man** pagina van **mkdir**

Wees er zeker van dat je in je home folder bent. Creëer dan een nieuwe folder met de **mkdir** opdracht.

```
cd
mkdir newdir
```

Creëer een aantal subfolders in *newdir*.

```
mkdir newdir/sub1 newdir/sub2 newdir/sub3
```

Probeer een folder te creëren in een folder waar je geen rechten hebt. Wat gebeurt er?

```
mkdir /etc/mydir
```



Oefeningen UNIX (toegangsrechten)



1. Verzeker je ervan dat je in je home folder bent. Voer volgende commando's uit (toegangsfouten mag je negeren):

```
cp -r /etc ~/etc
cd etc
ls -al
```

2. Let op de toegangsrechten toegewezen aan elk bestand. Kun je er uit afleiden:

- Welke bestanden folders zijn?
- Of een bestand uitvoerbaar is?
- Wie schrijfrechten heeft op elk bestand?
- Wie leesrechten heeft op elk bestand?
- Wie eigenaar is van het bestand?
- Tot welke groep de eigenaar behoort?
- Welke rechten de groep heeft?
- Welke rechten de gebruikers hebben?

3. Gebruik de **chmod** opdracht om de rechten te wijzigen op enkele bestanden. Lijst de bestanden uit **voor** en **na** elke wijziging. Ga na hoe de toegangsrechten op de bestanden zijn aangepast.

```
ls -l yp.conf
chmod o-r yp.conf
ls -l yp.conf
chmod ugo+wx yp.conf
ls -l yp.conf
chmod g-x yp.conf
ls -l yp.conf
```

4. Lees de **man** pagina van de **chown** opdracht. Dit programma dient om de eigenaar van een bestand te wijzigen. Probeer volgende opdracht.

```
chown root.root yp.conf
```

Waarom [lukt dit]/[lukt dit niet]?

Oefeningen UNIX (editors)

Deze oefeningen zullen je vertrouwd maken met de vi en pico editors

Een vi sessie starten

1. Ga naar de folder met oefenbestanden
`cd ~/etc`
2. Begin een **vi** sessie met het bestand *info-dir*
`vi info-dir`
3. Zodra het bestand op het scherm verschijnt, zul je een aantal dingen opmerken:
 - De bestandsnaam en bestandsgrootte verschijnen onderaan het scherm.
 - De cursor bevindt zich in de linker bovenhoek van het scherm.
 - Een volledig scherm tekst verschijnt, beginnend met de eerste regel in het bestand.



BELANGRIJK: De **vi** editor zal steeds in *command mode* starten. Wat je ook typt, het zal worden geïnterpreteerd als een commando en niet als in te voegen tekst.

Positionering van de cursor

1. Oefen de cursorverplaatsing voor één tegen tegelijk.

Pijlen	teken naar links, rechts, boven, beneden
backspace	één teken naar links
spatiebalk	één teken naar rechts
h	één teken naar links
l	één teken naar rechts
j	één teken naar beneden
k	één teken naar boven
2. Oefen de verplaatsing van de cursor woord per woord.

w	begin van een volgend woord
xw	begin van het x-de woord na de cursor
b	terug naar het vorige woord
xb	terug naar het x-de woord voor de cursor
e	einde van het volgende woord
xe	einde van de x-de woord na de cursor

3. Oefen de verplaatsing regel per regel.

+	begin van de volgende regel
-	begin van de vorige regel
0 (nul)	eerste kolom van de huidige regel
^	eerste teken van de huidige regel
\$	laatste teken van huidige regel

4. Oefen de verplaatsing blok per blok.

(begin van de zin
)	einde van de zin
{	begin van de paragraaf
}	einde van de paragraaf

5. Oefen de cursorverplaatsing per scherm.

CTRL-f	één scherm naar beneden
CTRL-b	één scherm naar boven
CTRL-d	half scherm naar beneden
CTRL-u	half scherm naar boven
H	bovenste regel van het scherm
M	middelste regel van het scherm
L	laatste regel van het scherm

6. Oefen de verplaatsing naar een specifieke regel in een bestand.

xG	x-de regel van het bestand
:x	x-de regel van het bestand
\$G	laatste regel van het bestand
1G	eerste regel van het bestand

Tekst invoegen

- De "andere" vi modus is de *insert modus*. In deze modus wordt de tekst die je typt geïnterpreteerd als tekst die moet worden toegevoegd aan het document.
- In de veronderstelling dat je nog steeds het bestand *info-dir* editeert, plaats je de cursor aan het begin van het bestand (**1G**). Plaats hem dan op de eerste letter van het woord *the*.
- Gebruik het "i" commando om je in de insert modus te plaatsen. Alles wat je nu typt wordt toegevoegd aan het bestand. Typ nu het volgende: "not ". Druk op de ESC-toets als je klaar bent, zo verlaat je de *insert modus*.
- Verplaats de cursor naar de laatste regel van de eerste paragraaf. Gebruik het commando "o" om een regel toe te voegen. Typ nu op deze regel "software". Druk op ESC om de *insert modus* te verlaten.

- De derde regel van de tweede paragraaf mist een woord. Verplaats je cursor naar het einde van die regel. Gebruik het "A" commando om het woord " **finally**" toe te voegen op het einde van de regel. Druk op ESC als je klaar bent.
- Wat doet het commando "a"?
- Bewaar de aangebrachte wijzigingen met de opdracht ":w". Merk op dat het een goede gewoonte is je aanpassingen regelmatig weg te schrijven.

Tekst verwijderen

- Ga met de cursor naar de laatste regel van de tweede paragraaf, op het einde van het woord <Return> (met de cursor op >). Typ een "x" om de > te verwijderen.
- Plaats de cursor op de eerste regel.. Geef het commando "D". Geef nu tweemaal het commando "dd". Wat is het verschil tussen deze twee commando's?
- Wat gebeurt er als je het commando "5dd" ingeeft?

Tekst wijzigen

- Typ het "R" commando in. Je zal nu zien dat je over tekst heen kan typen. Druk ESC om terug te gaan naar *commando modus*.
- Je kan regels aan elkaar koppelen met het "J" commando.

Regels kopiëren

- Ga op een willekeurige regel staan. Druk het commando "yy" in. Plaats nu de cursor op een andere plaats in het document. Geef nu het "p" commando. Je zal merken dat we de primitieve versie van *copy-paste* hebben nagebootst.
- Wat gebeurt er als we de handeling van het vorige punt herhalen, maar dan met het commando "3yy".
- Probeer uit te maken wat het verschil is tussen het "p" en het "P" commando (de buffer is nog intact, dus je kan gerust een aantal keren proberen).

Regels verplaatsen

- Gebruik op een willekeurige regel het commando "dd". Verplaats de cursor naar een andere plaats en plak de tekst met het "p" commando.
- Probeer het nu nog eens met het commando "5dd".

Zoeken/Vervangen

- Geef het commando `/Libc` om het woord **Libc** te zoeken in de tekst. De cursor zal geplaatst worden op de eerste plaats waar het woord voorkomt. Met het commando `/` zoek je naar de volgende verschijning van het ingegeven woord (we gaan dit niet doen omdat het woord ook maar één keer voorkomt).
- Ga naar het begin van de tekst en geef het commando `:%s/Libc/glibc/` in. Het eerste woord dat aan de beschrijving voldoet zal vervangen worden. Indien je het `%` teken niet bijvoegt zal er enkel op de huidige regel worden gezocht.
- In het begin van de tekst geef je het commando `:%s/the/blabla/gc`. Het programma gaat dan heel de tekst af om het woord te vervangen. Hierbij wordt er elke keer naar een bevestiging gevraagd. "y" is ja, "n" betekend nee, gebruik "a" om alles te wijzigen, met "q" stop je.

Sluiten

- Sluiten zonder bewaren: `:q!`
- Sluiten met bewaren: `:wq`



pico, een alternatief voor vi

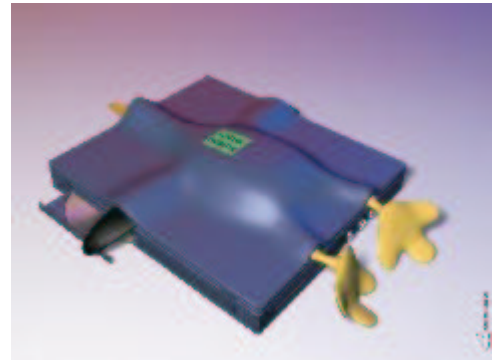
- Pico is vrij gemakkelijk in gebruik. Gebruik de onderste balk om je te assisteren in de oefening.
- Plaats je in de `~/etc` directory (indien dit niet het geval was)
- Open het **ipserv.conf** bestand.

```
pico ipserv.conf
```
- Verplaats je in het eerste scherm met de pijltjestoetsen. Probeer iets in te voeren. Je zal merken dat **pico** geen *commando modus* heeft zoals **vi**. Wat je typt wordt automatisch geïnterpreteerd als tekst en ingevoegd in het document. Verwijder dan hetgeen je hebt ingetypt.
- Probeer tekst te verwijderen, met de cursor in de tekst te verplaatsen en blanco regels toe te voegen.
- Probeer nu tekst te knippen en te plakken. Als oefening mag je nu zelf wat proberen.
- Indien je problemen hebt kan CTRL-G worden gebruikt om je bij te staan.
- Indien je klaar bent druk je op CTRL-X om te beëindigen.

Software installeren

RPM

RPM staat voor **RedHat Package Manager**. Het is een bestand waarin software zit die men kan installeren. Bij de installatie wordt er ook gecontroleerd op afhankelijkheden met andere RPM's (bijvoorbeeld een bepaalde RPM kan de functie van een andere RPM nodig hebben. De eerste RPM zal weigeren te installeren totdat de tweede beschikbaar is).



Er zijn twee soorten van RPM's: **binaire** en source RPM's.

- Binaire RPM's bevatten uitvoerbare bestanden. Deze zijn specifiek aan een bepaalde architectuur. Aan de bestandsnaam kan men zien voor welke architectuur deze RPM bestemd is (vb *naam-versie.i386.rpm* of *naam-versie.ppc.rpm*).

Installatie van zo'n RPM gebeurt met het commando:

```
rpm -Uvh naambestand.rpm
```

We doen eigenlijk een **upgrade**, maar indien het pakket nog niet bestaat wordt er overgegaan tot installatie

Nota: RPM's kunnen ook worden geïnstalleerd via de GUI. Het volstaat meestal om op het bestand te dubbelklikken.

- Source RPM's (SRPMS): dit zijn bestanden die de broncode bevatten. Om zo'n RPM's te installeren moet men een compiler ter beschikking hebben. Deze mogelijkheid bestaat om de gebruiker de kans te geven een zo geoptimaliseerd mogelijk programma te maken, of om architecturen te ondersteunen die niet vaak voorkomen. Met het commando "`rpm --rebuild naambestand.src.rpm`" zullen we het bestand uitpakken en compileren. Na het compileren vind je in "`/usr/src/redhat/RPMS/arch`" een **binaire** RPM die je kan installeren.

Je kan met het commando "`rpm -qa`" opvragen welke RPM pakketten er op het systeem zijn geïnstalleerd.

DEB

Debian maakt gebruik van een systeem gelijkaardig aan RPM. Dit zijn de **.deb* bestanden. Deze bestanden kunnen niet geïnstalleerd worden op een *RedHat* systeem.

Cursus Linux

Binary

Dit staat gelijk met een *self-extracting file*. Het bestand uitvoeren is meestal voldoende.

Nota: Vergeet niet het bestand na een download uitvoerbaar te maken met "`chmod +x bestand`".

TAR.GZ / TGZ

Deze bestanden zijn vergelijkbaar met een **.zip** bestand. Men kan deze uitpakken door het commando "`tar -zxvf naambestand.tar.gz`".

In een **tar.gz** bestand kunnen binaire bestanden zitten (die je uitpakt en uitvoert), maar meestal wordt dit formaat gebruikt om *broncode* in te bewaren.

Als alle standaarden zijn gevolgd in het pakket kan je met volgende commando's de zaak installeren (op voorwaarde dat je een compiler hebt geïnstalleerd):

- `./configure`
- `make`
- `make install`

De bestanden komen dan in **/usr/local/bin** terecht.

Bij pakketten die de standaarden niet volledig volgen kan je meestal de zaak compileren met een "make" en zal je ergens in de broncode-map wel een binary vinden (indien de compilatie gelukt is).

Opdracht:

Installeer de "**linuxconf**" software indien dit nog niet gebeurd is.

Sommige distributies (zoals Mandrake) installeren "**linuxconf**" automatisch. Als het programma nog niet geïnstalleerd is, zoek het dan eerst op de distributie CD's. Indien dit niet terug te vinden is op de CD's, download het programma van dit internet adres:

<http://www.solucorp.qc.ca/linuxconf>.

Gebruikersbeheer

Gebruikers

Linux heeft een gebruikersbeveiliging. Dit wil zeggen dat je bepaalde gebruikers meer rechten kan toekennen dan andere. Dit kunnen onder andere rechten zijn in verband met systeemadministratie of bestandstoegang.

De **root** gebruiker is de tegenhanger van de *Administrator*. Deze gebruiker kan alle bestanden lezen, wachtwoorden aanpassen,... (kortweg: alles). Het is afgeraden om als deze gebruiker te werken, vooral voor mensen met weinig UNIX ervaring (als men iets mis doet kan men het systeem om zeep helpen). Men kan ook gebruikers aanmaken die geen rechtstreekse toegang hebben tot het systeem (vb: enkel e-mail, inbeltoegang,...).

Groepen

In een groep kan je één of meerdere gebruikers bij elkaar plaatsen. Je gebruikt dit meestal om personen te groeperen die op een bepaalde items dezelfde toegang hebben. In een groep kan je in plaats van gebruikers ook andere groepen plaatsen (al dan niet gecombineerd met gebruikers).

Becoming God

Stel je voor dat je aan het werken bent als een gewone gebruiker en je wil een bewerking uitvoeren als **root** (enkel indien je het paswoord weet natuurlijk). Je kan via *remote-login* ook meestal niet inloggen als **root**.

Je kan dan tijdelijk **root** worden door het commando "**su**", gevolgd door het paswoord van **root**. Het is aan te raden het commando "**su -**" te gebruiken zodat je ook nieuwe omgevingsvariabelen van **root** laadt.

Een tweede mogelijkheid is het configureren van **sudo**. Hiermee kan je voor bepaalde gebruikers de toelating geven om bepaalde programma's te starten als *root*. De configuratie gebeurt in "*/etc/sudoers*" De gebruiker moet dan zijn commando starten als "*sudo programma*".

Bestand

De lijst van alle gebruikers zit in "*/etc/passwd*". Dezelfde lijst met geëncrypteerde paswoorden wordt in "*/etc/shadow*" bewaard. Het *shadow* bestand is geen bron van beveiligingslekken omdat het enkel leesbaar is door **root** (dus andere gebruikers kunnen deze niet lezen) én de paswoorden zijn versleuteld met een MD5 hash, dit wil zeggen dat een *brute force* attack (kraken) weinig uithaalt.

[Niet alle gebruikers in dit bestand kunnen aanloggen!](#)

Cursus Linux

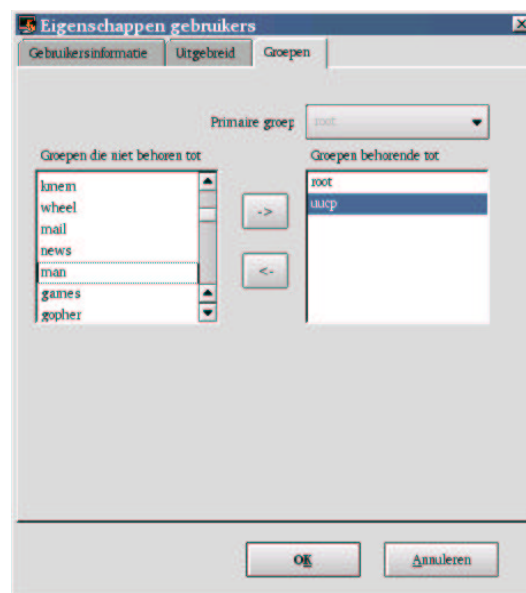
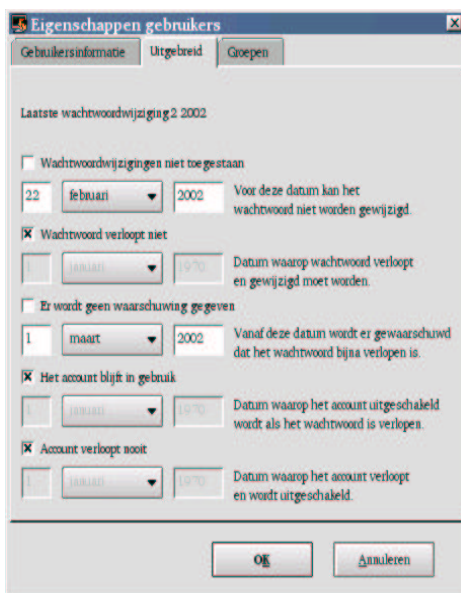
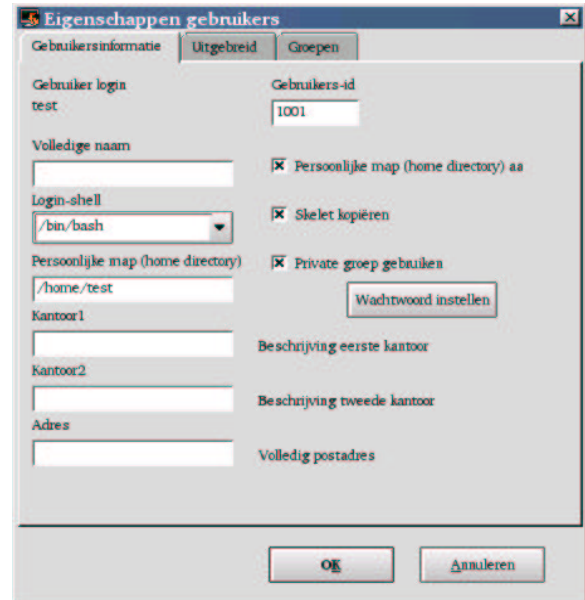
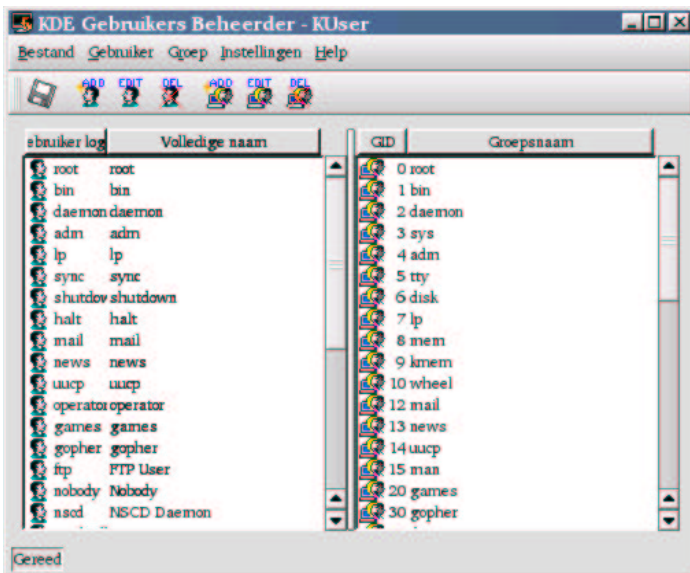
Aanmaken van gebruikers en groepen

Er zijn twee (drie) manieren om gebruikers aan te maken/wijzigen.

Grafische manier

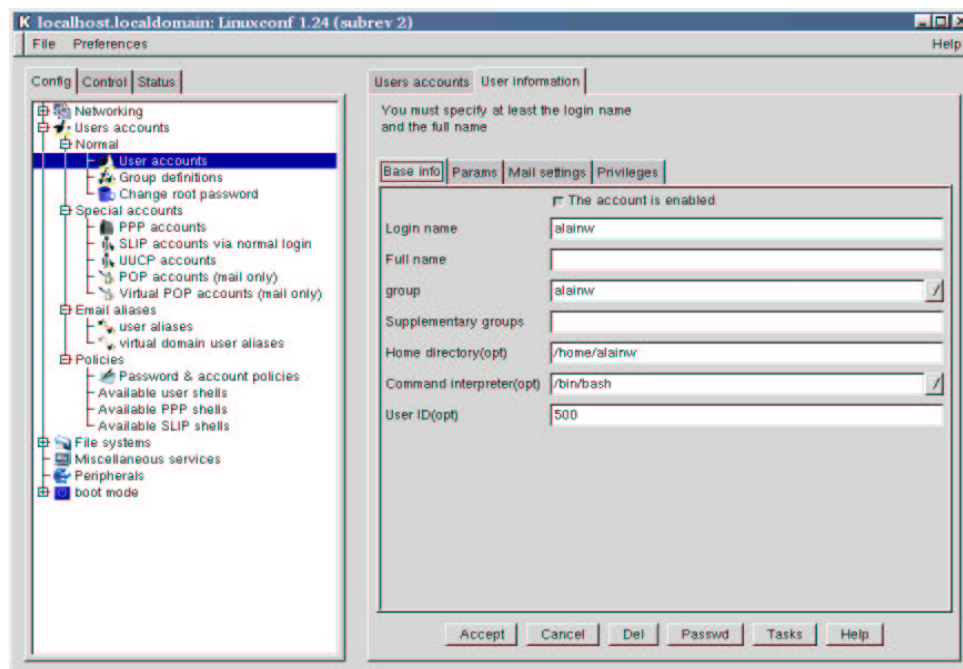
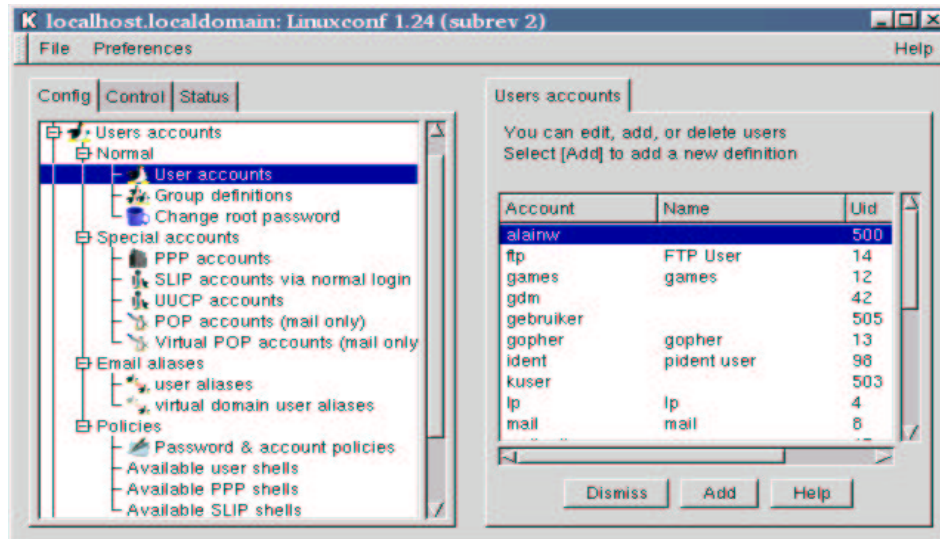
Linuxdistributies bevatten allerlei tools om dit te doen. Eén voorbeeld is de KDE-gebruiksbeheerder.

Hieronder enkele screenshots (het gebruik wijst zichzelf uit):



Cursus Linux

Je kan ook met **linuxconf** de gebruikers beheren. Deze biedt nog de extra optie om speciale gebruikers aan te maken (die bijvoorbeeld niet mogen aanloggen zoals mail- en inbelgebruikers,...). Het gebruik wijst zich eveneens vanzelf uit.



Opdracht: probeer eens een gebruiker aan te maken, en probeer dan in te loggen als die gebruiker (ter controle).

Commando's

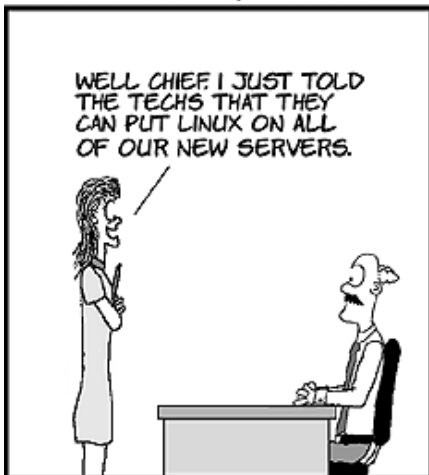
Je kan in plaats van de GUI (**linuxconf** kan trouwens ook in *tekstmode* starten) ook commando's gebruiken. Specifiek voor het gebruik verwijzen we naar de **man-pages**.

- **useradd**: aanmaken van een gebruiker
- **userdel**: verwijderen van een gebruiker
- **usermod**: wijzigen gegevens van een gebruiker
- **groupadd**: aanmaken van een groep
- **groupdel**: verwijderen van een groep
- **groupmod**: wijzigen van een groep

Met het "**passwd**" commando wijzig je het paswoord.

Opdracht: Probeer eens een gebruiker aan te maken met deze methode. Log daarna eens in als deze gebruiker (ter controle).

USER FRIENDLY by Illiad



<http://www.userfriendly.org>

Procesbeheer

Omdat Linux een **multitasking** OS met *memory protection* is kunnen we processen beheren. Er zijn *tools* voorhanden om de lijst van alle processen op te vragen en processen te stoppen.

De console

In de console kunnen we de processen beheren aan de hand van enkele simpele commando's.

- Het opvragen van een lijst van alle processen kunnen we met `ps -ef` (of `ps -auxw` op oudere systemen)

Je krijgt dan een lijst in de vorm van (ingekort):

```
root      1      0  0 11:03 ?        00:00:04 init [3]
root      2      1  0 11:03 ?        00:00:00 [keventd]
root      3      1  0 11:03 ?        00:00:00 [kapm-idled]
root      4      1  0 11:03 ?        00:00:00 [kswapd]
root      5      1  0 11:03 ?        00:00:00 [kreclaimd]
root      6      1  0 11:03 ?        00:00:00 [bdflush]
root      7      1  0 11:03 ?        00:00:00 [kupdated]
root      8      1  0 11:03 ?        00:00:00 [mdrecoveryd]
root     73      1  0 11:03 ?        00:00:00 [khubd]
root    396      1  0 11:03 ?        00:00:00 syslogd -m 0
root    401      1  0 11:03 ?        00:00:00 klogd -2
root    467      1  0 11:03 ?        00:00:00 /sbin/cardmgr
root    556      1  0 11:03 ?        00:00:00 /usr/sbin/apmd -p 10 -w 5 -W -P
root    682      1  0 11:03 ?        00:00:00 /usr/sbin/automount --timeout 60
root    697      1  0 11:03 ?        00:00:00 /usr/sbin/sshd
root    946      1  0 11:03 tty6      00:00:00 /sbin/mingetty tty6
root    949    941  0 11:04 tty1      00:00:00 -bash
root    983    949  0 11:04 tty1      00:00:00 /bin/sh /usr/X11R6/bin/startx
root    988    983  0 11:04 tty1      00:00:00 xinit /etc/X11/xinit/xinitrc --
root    989    988  1 11:04 ?        00:00:16 /etc/X11/X :0 -auth /
root    993    988  0 11:04 tty1      00:00:00 kmsserver --restore
root   1042      1  0 11:04 ?        00:00:00 kdeinit: dcopserver --nosid
root   1044      1  0 11:04 ?        00:00:00 kdeinit: klauncher
root   1046      1  0 11:04 ?        00:00:00 kdeinit: kded
root   1049      1  0 11:04 ?        00:00:01 artsd -F 10 -S 4096
root   1054      1  0 11:04 ?        00:00:00 kdeinit: kxmlrpcd
root   1063    944  0 11:04 tty4      00:00:00 -bash
  1         2         3         4         5         6         7
```

1 = De gebruiker die het proces heeft opgestart

2 = Het process ID

3 = Het proces waar het huidig proces afhankelijk van is (1 = geen afhankelijkheid)

4 = Systeemtijd wanneer het proces is opgestart

5 = Op welk scherm het proces draai (? = geen scherm)

6 = Systeemtijd dat het proces al heeft gedraaid

7 = Naam van het proces

- Het stoppen van een proces doe je met de **kill/killall** opdracht.

Syntax = `kill procesID`

Vb: `kill 1050`

Je kan ook de **killall** opdracht gebruiken. Je verwijdert dan de opdracht aan de hand van de naam in plaats van een proces ID. Indien meerdere processen aan jouw beschrijving voldoen worden deze ook verwijderd.

Syntax = `killall procesnaam`

Vb: `killall soffice.bin`

Je kan enkel processen afsluiten waarvan je eigenaar bent (tenzij je **root** bent)

Indien **kill/killall** wordt gebruikt zonder extra opties wordt er *vriendelijk* aan het proces gevraagd om op te houden en proper af te sluiten. Als een proces *gecrasht* is kan dit natuurlijk niet meer. We kunnen **kill/killall** starten met de optie "-9". Dit zorgt ervoor dat de kernel het volledige proces opruimt en het geheugen vrijgeeft (zonder dat het *memory leaks* veroorzaakt).

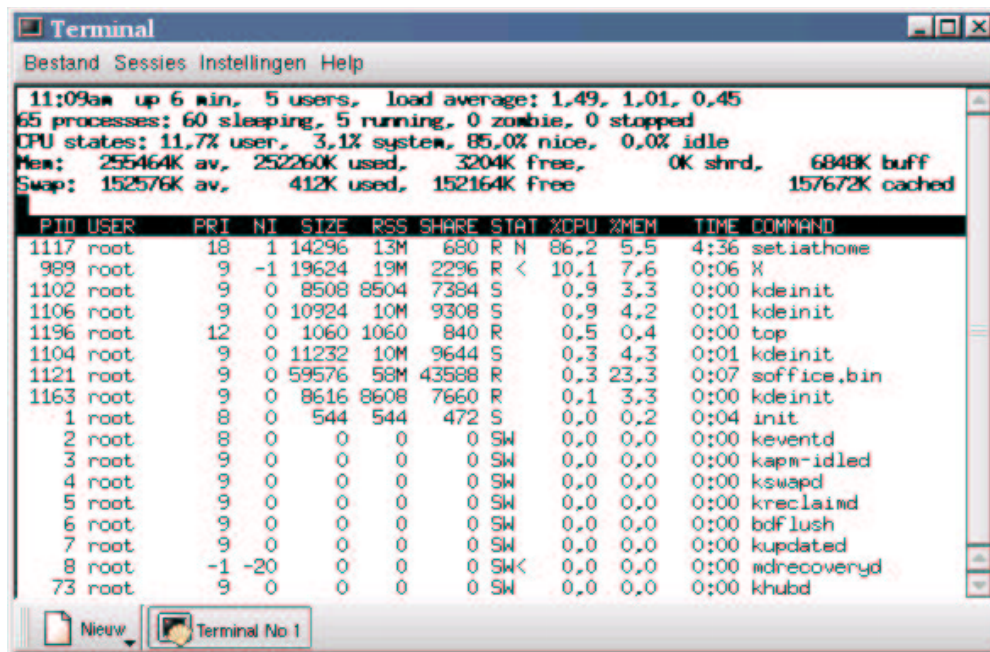
Vb: `kill -9 1050`

Met **kill/killall** kunnen we ook nog andere berichten sturen naar processen. Zo zullen serverprocessen op een **SIGUSR** hun logs roteren of archiveren.

Vb: `kill -SIGUSR 1050`

De **kill/killall** stuurt het bericht **SIGTERM**. De optie "-9" stuurt een **SIGKILL**.

- Om processen in het oog te houden kan men gebruik maken van het **top** commando.



```
Terminal
Bestand Sessies Instellingen Help
11:09am up 6 min, 5 users, load average: 1.49, 1.01, 0.45
65 processes: 60 sleeping, 5 running, 0 zombie, 0 stopped
CPU states: 11.7% user, 3.1% system, 85.0% nice, 0.0% idle
Mem: 255464K av, 252260K used, 3204K free, 0K shrd, 6848K buff
Swap: 152576K av, 412K used, 152164K free

  PID USER   PRI  NI  SIZE  RSS  SHARE STAT %CPU %MEM  TIME COMMAND
 1117 root    18   1 14296  13M  680 R N   86.2  5.5   4:36 setiathome
   989 root     9  -1 19624  19M 2296 R <  10.1  7.6   0:06 X
  1102 root     9   0  8508  8504 7384 S     0.9  3.3   0:00 kdeinit
  1106 root     9   0 10924  10M  9308 S     0.9  4.2   0:01 kdeinit
  1196 root    12   0  1060  1060  840 R     0.5  0.4   0:00 top
  1104 root     9   0 11232  10M  9644 S     0.3  4.3   0:01 kdeinit
  1121 root     9   0 59576  58M 43588 R     0.3 23.3   0:07 soffice.bin
  1163 root     9   0  8616  8608 7660 R     0.1  3.3   0:00 kdeinit
     1 root     8   0   544   544  472 S     0.0  0.2   0:04 init
     2 root     8   0     0     0     0 SW     0.0  0.0   0:00 keventd
     3 root     9   0     0     0     0 SW     0.0  0.0   0:00 kpm-idled
     4 root     9   0     0     0     0 SW     0.0  0.0   0:00 kswapd
     5 root     9   0     0     0     0 SW     0.0  0.0   0:00 kreclaimd
     6 root     9   0     0     0     0 SW     0.0  0.0   0:00 bdflush
     7 root     9   0     0     0     0 SW     0.0  0.0   0:00 kupdated
     8 root    -1 -20     0     0     0 SW<    0.0  0.0   0:00 mdrecoveryd
    73 root     9   0     0     0     0 SW     0.0  0.0   0:00 khubd
```

Je vindt er interessante informatie zoals hoelang het systeem draait, hoeveel bezetting, hoeveel processen er draaien en hoeveel geheugen de processen innemen.

De GUI

In het GUI systeem zijn er ook enkele *tools* om processen te beheren. De meest gekende is **gtop** of **ktop** (afhankelijk van welke desktop je hebt geïnstalleerd).

De werking is gelijkaardig aan **top**.

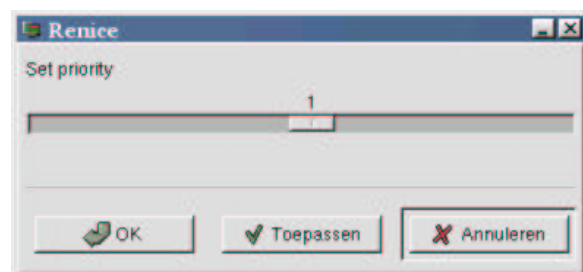
PID	User	Pri	Size	Resident	Stat	CPU	MEM	Time	Cmd
1117	root	20	15324	15324	R N	96.0	5.9	6:09m	./setiathome
989	root	9	19800	19800	S <	0.6	4.1	9.50s	/etc/X11/X
1121	root	9	59576	59576	S	0.0	23.3	7.93s	/usr/local/office52/
1	root	8	544	544	S	0.0	0.2	4.77s	init [3]
1202	root	9	18032	18032	S	0.0	7.0	1.68s	gimp
1106	root	9	10924	10920	S	0.6	4.2	1.31s	kdeinit: kicker
1104	root	9	11232	11228	S	0.0	4.3	1.30s	kdeinit: kdesktop
904	xfs	9	4376	4376	S	0.0	1.7	1.26s	xfs
1049	root	9	3024	3024	S	0.0	1.1	1.24s	artsd
1211	root	14	4724	4724	R	1.9	1.8	1.24s	gtop
1102	root	9	8524	8520	S	0.6	3.3	1.11s	kdeinit: kwin
1163	root	9	8616	8608	S	0.0	3.3	0.85s	kdeinit: konsole -ic
1140	root	9	8496	8488	S	0.0	3.3	0.46s	kdeinit: konsole -ic
1110	root	9	8568	8564	S	0.0	3.3	0.44s	kdeinit: klipper -ic
1114	root	9	7760	7752	S	0.0	3.0	0.37s	kdeinit: kwrite
1203	root	9	2904	2904	S	0.0	1.1	0.36s	/usr/lib/gimp/1.2/pl
1112	root	9	7344	7332	S	0.0	2.8	0.35s	kdeinit: khotkeys
2	root	8	0	0	SM	0.0	0.0	0.28s	keventd

Indien je met de muis over een proces beweegt en de rechter muistoets aanklikt krijg je meerdere opties. Je kan oa. Het proces afsluiten of een ander bericht sturen.

Je kan ook de prioriteit van een proces bijstellen met **Renice**.

Een waarde dicht bij **-20** is een *hoge prioriteit*. Waarden dicht bij **+20** veroorzaken een *lagere prioriteit*.

Deze functionaliteit kan je ook via de commando's `nice` en `renice` beheren.



Het onderliggend systeem

Init

Wanneer een Linux systeem opstart worden er een aantal scripts uitgevoerd. Deze scripts starten een aantal processen (*daemons*) op.



Een Linux heeft verschillende **runlevels**. In iedere runlevel kan de gebruiker/administrator zelfs definiëren welke processen er mogen gestart worden. Het standaard aantal **runlevels** en hun functie kan bij iedere distributie verschillen. Bij *RedHat* geïnspireerde distro's worden de runlevels als volgt gebruikt:

- 1 = Single User (Een enkele gebruiker kan inloggen. Er is geen netwerkconnectiviteit)
- 3 = Multi-User Text (Normale operatiemodus)
- 5 = Multi-User X (idem aan 3, behalve dat de X-server continue draait)
- 6 = Shutdown (alle processen stoppen)

De administrator kan het systeem van runlevel veranderen door het commando:

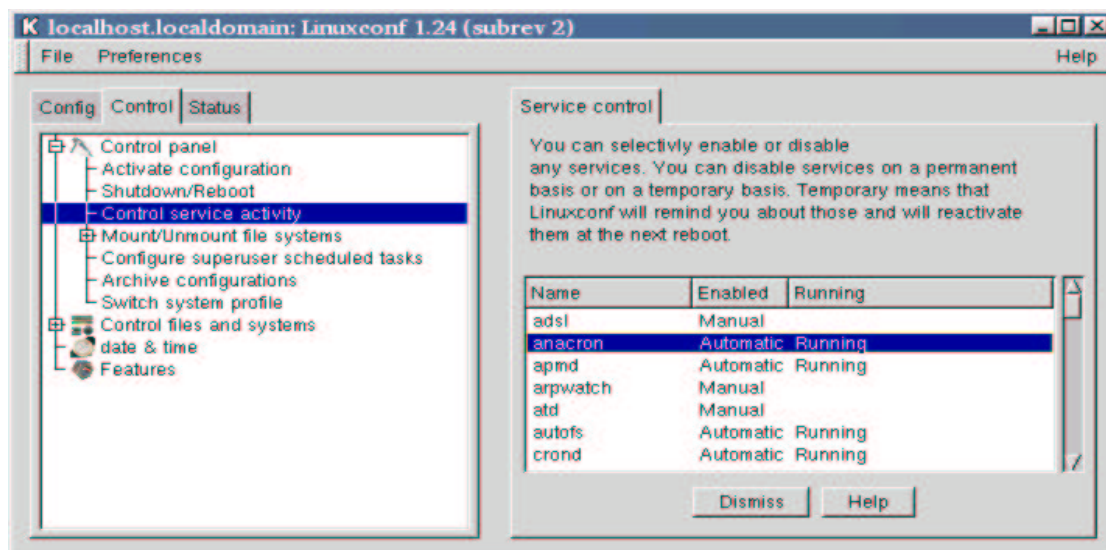
```
init x (x = gewenste runlevel)
```

Men kan individueel daemons starten en stoppen door scripts in `"/etc/init.d"`

Syntax = `/etc/init.d/naamscript {start | stop | restart}`

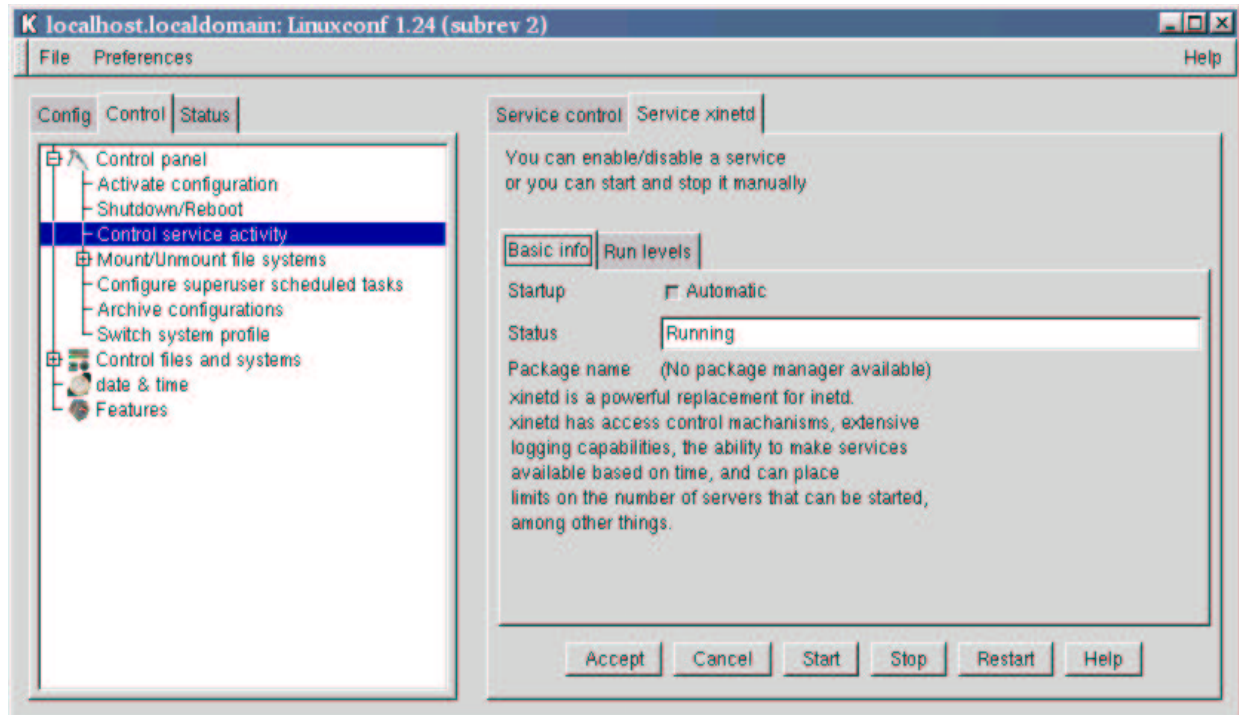
Het bestand `/etc/rc.d/rc.local` wordt bij iedere start uitgevoerd.

De keuze welke daemons mogen draaien in welke runlevel kan aangepast worden in *Linuxconf* (screenshot van de grafische versie). Kies het *tabblad* "Control", in de lijst klik je op "Control Service Activity".



Cursus Linux

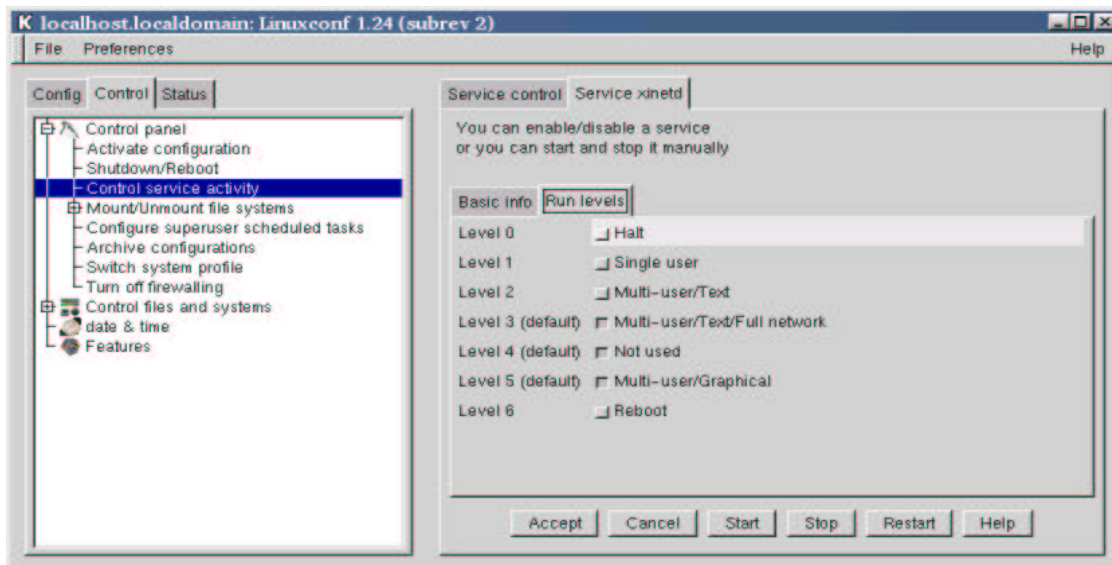
Klik op de *daemon* waar je aanpassingen voor wil maken.



Met "**Automatic**" kan je aangeven of de *daemon* moet gestart worden.

Met de *knoppen Start/Stop/Restart* kan je de *daemon* beheren.

Als je op het *tabblad "Runlevels"* klikt kan je instellen in welke runlevels het programma wordt gestart.



Cron

Cron is één van die *daemons* die nodig zijn voor de correcte werking van het systeem. Cron dient om op regelmatige tijdstippen programma's of scripts uit te voeren. Dit kan ingesteld worden in `/etc/crontab`.

Als we kijken hoe een normaal **crontab** bestand eruit ziet krijgen we dit:

```
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
 1  2  3  4  5  6      7
```

Kort uitgelegd wat dit doet:

- Ieder uur (x:01) worden alle scripts in `/etc/cron.hourly` uitgevoerd
- Iedere dag om 4:02 worden alle scripts in `/etc/cron.daily` uitgevoerd
- Iedere zondag om 4u22 worden alle scripts in `/etc/cron.weekly` uitgevoerd
- Iedere eerste dag van de maand wordt om 4u42 alle scripts in `/etc/cron.monthly` uitgevoerd.

Dit zijn reeds klaargemaakte *entry's*. Je kan zelf ook programma's starten van hieruit (je hoeft niet met directories te werken).

We overlopen de betekenis van de verschillende kolommen:

1 = minuut (0-59)

2 = uur (1-23)

3 = dag van de maand (1-31)

4 = maand (1-12)

5 = dag van de week (0-7, 0 = Zondag)

6 = gebruiker die het uitvoert (niet nodig, enkel geldig voor **root**)

7 = programma dat uitgevoerd moet worden

Een * staat voor *altijd* of *irrelevant*.

Meestal zullen we de crontab van de **root** gebruiker aanpassen, maar het is mogelijk om meerdere gebruikers een eigen *crontab* te geven. Log in als die bepaalde gebruiker en typ het commando `crontab -e`. Je krijgt dan een **vi** venster waar je de **crontab** in kwijt kunt.

Kernel Drivers

De drivers voor hardware worden meestal met de kernel geleverd. De drivers kan men meestal vinden in `/lib/modules/2.4.x-y/kernel/drivers`. Men vindt hier *.o bestanden die men *at runtime* kan laden.

In de meeste gevallen worden drivers automatisch geladen. Indien het wel nodig blijkt te zijn om zelf iets te veranderen, zijn er enkele tools beschikbaar om wijzigingen aan te brengen.

Om een lijst op te vragen van alle geladen drivers typ je "lsmod". Je krijgt dan een lijst die hierop lijkt:

```
Module          Size  Used by
ltmodem         361472  0
xircom_cb        5936    1
ds               7280    2  [cb_enabler]
yenta_socket    11440    2
pcmcia_core     43072    0  [cb_enabler ds yenta_socket]
ipchains        38976    0  (unused)
usb-uhci        20720    0  (unused)
usbcore         49664    1  [usb-uhci]
```

Het laden van een driver kan door het commando "modprobe *modulenaam*".

Vb: modprobe ne2k-pci

Je kan ook het commando "insmod" gebruiken, maar dan is er geen hardware-probing.

Om te weten te komen wat voor hardware in je PC zit gebruik je het commando "lspci". Deze lijst kan ook opgevraagd worden in de GUI met configuratiemanagers. Voor een gedetailleerde lijst heb je ook nog de commando's "lspci -v" en "lspci -vv".

Indien je een driver wil uitschakelen, dan kan dit door het "rmmod" commando. Hou er wel rekening mee dat je enkel drivers kan *unloaden* indien deze niet in gebruik zijn.

Vb: rmmod ipchains

Logging

Logbestanden worden weggeschreven in `/var/log` (of onderliggende mappen).

Op *RedHat* gebaseerde distributies kan je in `/etc/logrotate.conf` volgende instellingen wijzigen:

- Hoeveel keer per week/dag/maand de logbestanden *geroteerd* moeten worden
- Hoelang de bestanden moeten worden bijgehouden
- Het compresseren van de oude logbestanden

Bij het *roteren* van het logbestand wordt het oude logbestand "*logbestand*" hernoemd naar "*logbestand.1*". Het oude "*logbestand.1*" wordt "*logbestand.2*" enz. Er wordt een leeg "*logbestand*" aangemaakt.

Specifiekere configuratiemogelijkheden vind je in `/etc/logrotate.d`.

In *SuSE* wordt er gebruikt gemaakt van een `/etc/logfiles` waarin beschreven staat welke logbestanden in het oog worden gehouden en hoe groot ze maximum mogen worden. Een voorbeeld uit zo'n `/etc/logfiles`.

```
# File                max size  mode  ownership  service
#                    (reload if changed)
/var/log/log.nmb      +1024k    644   root.root
/var/log/log.smb      +1024k    644   root.root
/var/log/maillog      +1024k    640   root.root
/var/log/ntp          +2048k    640   root.root
/var/log/pop3svr.log  +1024k    640   root.root
/var/log/postgresql   +1024k    644   postgres.daemon
```

Bij het overschrijden van de aangegeven limiet wordt het logbestand gearhiveerd als `logbestand-datum.gz`. In `/etc/rc.config` staat aangegeven hoe lang de bestanden bewaard moeten worden.

Geluid

Indien een geluidskaart is geïnstalleerd bestaat er een `/dev/dsp` naar waar je geluid kunt naartoe sturen. Meestal ga je dit zelf niet doen, maar zijn er programma's die het voor jou doen.

Voor *RedHat* distributies moet je `sndconfig` draaien om jouw geluidskaart te installeren. Andere distributies doen dit meestal bij de installatie.

Jammer genoeg kan je niet meerdere *geluidsstreams* naar `/dev/dsp` sturen zodat er slechts één programma tegelijkertijd de geluidkaart kan gebruiken. Dit is ook zo op andere systemen, dit is nu eenmaal een tekortkoming in de geluidskaarten (Moderne *multithreaded* geluidskaarten kunnen dit wel. Linux maakt hier dan ook gebruik van).

Om deze tekortkoming weg te werken bestaan er *sounddaemons*. Daarnaast kan men op deze manier de geluidsstreams bewerken voor ze de geluidskaart bereiken.

Deze *sounddaemons* werden geschreven door de makers van GUI-desktops. Zo heeft KDE *artsd*, terwijl GNOME gebruik maakt van *eSound*. Welke er wordt gebruikt, maakt niet uit (de configuratie gebeurt ook automatisch). Programma's schrijven hun geluid niet meer weg naar `/dev/dsp`, maar naar de *geluidsdaemon*. Die *geluidsdaemon* kan geluidsstromen van verschillende programma's tegelijk ontvangen en gaat deze *mixen*. Op die manier lijkt het dat meerdere programma's tegelijkertijd de geluidskaart kunnen aanspreken.

Printen

De **lpd** daemon dient om printopdrachten in een wachtrij te zetten. Het is nodig om deze daemon te draaien indien men wil printen. Deze daemon gebruikt `/etc/printcap` voor de lijst van de printers en welke filters/drivers er gebruikt moeten worden. Je gaat `/etc/printcap` niet meer manueel configureren, daarvoor gebruik je een GUI-tool dat in *RedHat* "**printtool**" heet.

Het afprinten gebeurt **altijd** in *postscript*. De omzetting naar een commandoset voor de printer gebeurt pas op het laatste moment.

Applicatie (PS) ↗ LPD (server) ↗ GhostScript (PS naar printertaal) ↗ Printer

Je kan ook iets printen met het commando:

```
lpr naambestand.ps
```

Je hebt altijd een printerserver nodig om te kunnen afprinten. Alle programma's onder UNIX sturen bij het printen een postscript bestand naar de printserver. Deze printservers worden doorgaans gebruikt om locale printjobs uit te voeren, maar de instellingen kunnen aangepast worden zodat het systeem ook printjobs over het netwerk kan ontvangen.



Er bestaan drie printerservers:

- **Berkeley LPR:** De originele (en oudste) printer daemon uit UNIX. Deze daemon wordt ook nog gebruikt in de commerciële Unices en oudere Linux distributies.
- **LPR-NG:** Dit is de nieuwere versie van de *Berkeley LPR* (*NG* staat voor *New Generation*). De broncode is volledig herschreven om tekortkomingen uit de originele *LPR* weg te werken. Default wordt er bij RedHat *LPR-NG* gebruikt.
- **CUPS:** Common Unix Printing System. *CUPS* is een printerserver waarvan de structuur verschilt van de *LPR* systeem. Het print systeem is een nieuw idee en steunt op het *Internet Printing Protocol* (maar het ondersteunt natuurlijk ook nog altijd het originele *LPR* protocol). Momenteel leveren distributies zoals SuSE en Mandrake standaard deze printerserver.

IP

Wat is IP?

IP is een netwerkprotocol dat onder andere gebruikt wordt op het internet. IP laat toe om meerdere computers met elkaar te laten communiceren. De naam IP staat voor "Internet Protocol". Meestal maken we gebruik van een ethernet netwerk, maar IP kan ook gebruikt worden over andere media zoals bijvoorbeeld ADSL, kabel, dial-up/telefoon,...



Kenmerken

IP is vooral ontworpen voor grote netwerken, maar het wordt tegenwoordig ook gebruikt op kleinere netwerken omdat het dé standaard is geworden in netwerkcommunicatie. Andere protocollen zijn bijvoorbeeld IPX/SPX van Novell en NETBUI van Microsoft. Zowel Novell als Microsoft ondersteunen nu IP als standaard.

Eén van de grootste voordelen is dat je met IP gebruik kan maken van *routers* zodat je meerdere netwerken aan mekaar kan koppelen.

Een groot nadeel van IP is dat elk station (computer, server,...) een IP-adres moet hebben. Deze moet dan manueel toegekend worden.

Adressen

Een IP-adres bestaat uit vier getallen, ieder getal heeft een waarde van 0 tot 255. Zo is **10.1.1.30** een voorbeeld van zo'n adres. Het adres moet uniek zijn om vanaf een andere locatie het station met dit adres te kunnen aanspreken.

Indien je een computer op het internet wil aansluiten moet je een adres aanvragen bij een officiële instantie (voor Europa is dit RIPE www.ripe.net). Een internetprovider heeft dit voor jouw gedaan en "leent" je tijdelijk een IP-adres.

Er zijn drie reeksen van adressen die je vrij mag gebruiken op interne netwerken. De adressen worden niet gebruikt op het internet en het is door providers overeen gekomen dat deze ook niet worden *gerout*.

- 10.x.x.x
- 172.16-31.x.x
- 192.168.x.x

Volgende adressen worden ook niet gebruikt op het internet, maar het wordt afgeraden om toe te passen:

- 169.254.x.x (Microsoft Autoaddressing, officieel niet herkent)
- 127.x.x.x (niet gebruiken! Deze is gereserveerd voor systeemdoeleinden)

subnet mask

Bij het IP adres kan je ook een **subnet mask** instellen. Deze kan er bijvoorbeeld uitzien als "255.255.255.0".

Het subnet mask dient om uit te maken welk gedeelte van het IP-adres toebehoort tot het adres van het netwerk of hosts. Dit heeft belang als men wil *routen*.

Het gedeelte met de "255" duidt aan dat dit om het netwerkgedeelte gaat.

Bijvoorbeeld: 192.168.10.1 met subnet 255.255.0.0

```

                ↓
            192.168.010.001
            255.255.000.000
192.168 is het netwerkgedeelte
10.1    is het adres van de host
```

Alle computers met een gelijk netwerknummer kunnen met mekaar communiceren zonder bijkomende configuratie. Indien de *host* met een andere *host* wil communiceren dat een ander netwerknummer heeft, dan zal de *host* die de communicatie opzet gebruik maken van een *default gateway*.

De *default gateway* wordt door de computer gebruikt als *router*. Alle netwerkpakketjes die de locale host niet kan thuisbrengen worden naar de *default gateway* gestuurd.

Speciale adressen

Indien het host-gedeelte van het IP-adres "0" is, dan wordt hiermee het *netwerkadres* adres bedoeld. Bijvoorbeeld: **192.168.10.0**. Dit adres is niet bruikbaar voor computers. Het adres wordt wel gebruikt bij het aangeven van adresgroepen.

Indien het host-gedeelte "255" is, dan staat dit voor een *broadcast* adres. Dit adres kan men dan gebruiken om alle computers in het netwerk aan te spreken. Een voorbeeld is **192.168.10.255**. Alle andere adressen daartussen kunnen gebruikt worden door computers/toestellen.

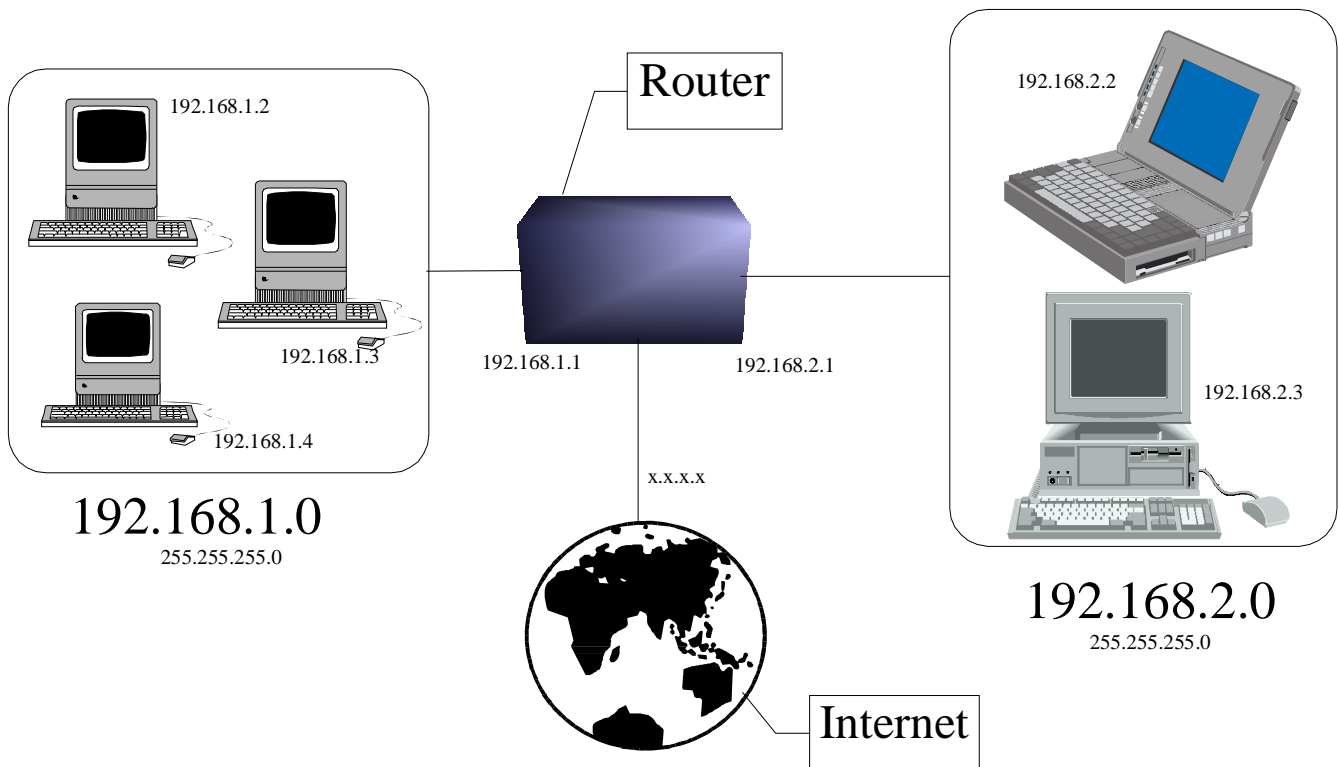
De locale computer heeft altijd **127.0.0.1** ingesteld als adres. Dit dient enkel voor communicatie tussen twee of meerdere programma's op dezelfde computer. Het is ook afgeraden om adressen te gebruiken in de **127.x.x.x range**.

Routing / Default gateway

Een *router* is een toestel/computer dat meerdere netwerken met elkaar verbindt. De router heeft, door meerdere verbindingen, kennis van de weg dat een bepaald IP-pakket moet afleggen om aan zijn bestemming te geraken.

Men kan routers kopen van gespecialiseerde fabrikanten (zoals Cisco), maar je kan er zelf ook één bouwen door een PC te voorzien van twee of meerdere netwerkkaarten.

Grafisch kan je dit zo voorstellen:



Sub-protocollen

Binnen IP heb je een aantal sub-protocollen zoals onder andere:

- ICMP (oa. voor ping)
- IGMP
- ARP (opzoeken van ip-adressen)
- TCP
- UDP
- ...

De meest belangrijke zijn **UDP** en **TCP**.

TCP is gelijkaardig aan UDP op één detail na. TCP maakt gebruik van *connections*, dit wil zeggen dat er tussen twee *hosts* eerst een vaste verbinding wordt onderhandeld. Bij TCP gaat de ontvangende computer ook telkens bevestigingen sturen dat het pakketje goed is toegekomen.

In het geval van UDP wordt er data verstuurd. De zendende computer weet niet of de data ooit is toegekomen.

UDP is uiteraard dan sneller dan TCP, maar omdat het niet controleerbaar is of data toegekomen is kan dit niet gebruikt worden om belangrijke data te sturen. Het toepassingsgebied van UDP is onder andere streaming (RealVideo, webcam, internetradio, ...).

Zowel TCP als UDP maken gebruik van *poorten*.

TCP & UDP poorten

Er zijn voor ieder subprotocol 65536 poorten voorzien. Een poort is een soort van adres waar een bepaalde applicatie achter draait. Er kan niet meer dan één applicatie een bepaalde poort bezetten.

Poorten onder de 1024 zijn gereserveerd voor servers (op hogere poorten kan je ook servers draaien, maar dit is niet vastgelegd). Zo zijn er een aantal poorten vastgelegd waar men vooraf van weet waar men uitkomt op een server. Enkele voorbeelden zijn:

- 21 ftp
- 22 secure shell
- 23 telnet
- 25 mailservers
- 80 webserver
- 110 POP3
- 137-139 Microsoft Sharing
- ...

ping

Indien je een controle wil uitvoeren om te weten of een bepaalde computer bereikbaar is kan je gebruik maken van het **ping** programma. De computer aan de andere kant zal, indien mogelijk, antwoorden. Dit programma is vooral handig bij het onderzoeken van netwerkproblemen.

Syntax:

```
ping x.x.x.x      (x.x.x.x is het IP-adres van de ontvanger)
```

Uitvoer:

```
PING 192.168.20.1 (192.168.20.1) from 192.168.20.1 : 56(84)
bytes of data.
Warning: time of day goes back, taking countermeasures.
64 bytes from 192.168.20.1: icmp_seq=0 ttl=255 time=367 usec
64 bytes from 192.168.20.1: icmp_seq=1 ttl=255 time=71 usec
64 bytes from 192.168.20.1: icmp_seq=2 ttl=255 time=64 usec
64 bytes from 192.168.20.1: icmp_seq=3 ttl=255 time=73 usec
64 bytes from 192.168.20.1: icmp_seq=4 ttl=255 time=73 usec
64 bytes from 192.168.20.1: icmp_seq=5 ttl=255 time=74 usec
< CTRL-C om te onderbreken >
--- 192.168.20.1 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.064/0.120/0.367/0.110 ms
```

DHCP

Bij IP moet je in principe iedere *host* een uniek IP-adres geven. Dit is nogal omslachtig omdat de administratie en de bijhorende moeilijkheden nogal omvangrijk zijn. Zo kan je de werking van een netwerk verstoren indien je twee of meerdere computers hebt met hetzelfde IP-adres.

Daarom is DHCP (Dynamic Host Configuration Protocol) uitgevonden.

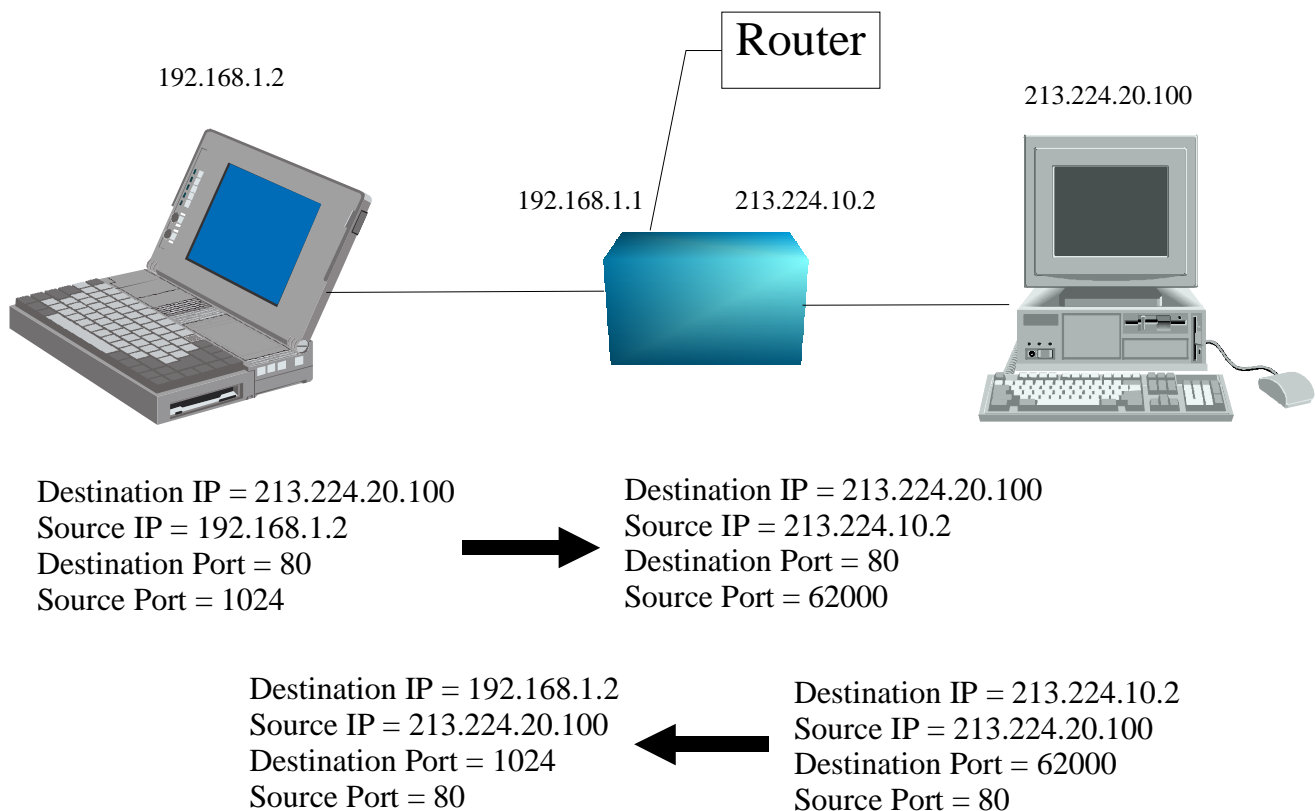
De computers moeten eerst worden ingesteld om het DHCP-protocol te gebruiken. Ergens in het netwerk moet er dan een DHCP server staan. Iedere keer dat de PC opstart gaat deze een IP-adres vragen aan de DHCP server. Deze geeft dan een adres uit de reeks van beschikbare adressen. De DHCP server houdt dan een database bij welke IP-adressen in gebruik zijn. Bij het uitzetten van de computer wordt het adres bij de DHCP server terug vrijgegeven zodat er een andere computer gebruik kan maken van dit adres. DHCP werkt enkel binnen één segment van een netwerk (dit wil zeggen dat er geen adressen kunnen gegeven worden aan computers achter een router).

NAT

NAT staat voor *Network Address Translator*. Dit maakt het mogelijk om een netwerk met *private* IP-adressen (192.168.x.x, 10.x.x.x.x,...) toch een **volwaardige** toegang te geven tot het internet. In de linuxwereld wordt er in plaats van *NAT* ook wel eens gesproken van *masquerading*.

NAT kan toegepast worden door een router die een internetverbinding heeft. Het verschil tussen NAT en gewone *routing* is dat bij NAT het IP-adres van de oorspronkelijke computer wordt omgevormd zodat het lijkt dat het pakketje afkomstig is van de router. NAT wordt vooral gebruikt om een internettoegang te delen, maar de toepassing ervan is ook mogelijk zonder internet (alhoewel men dan kan twijfelen of het nog nodig is).

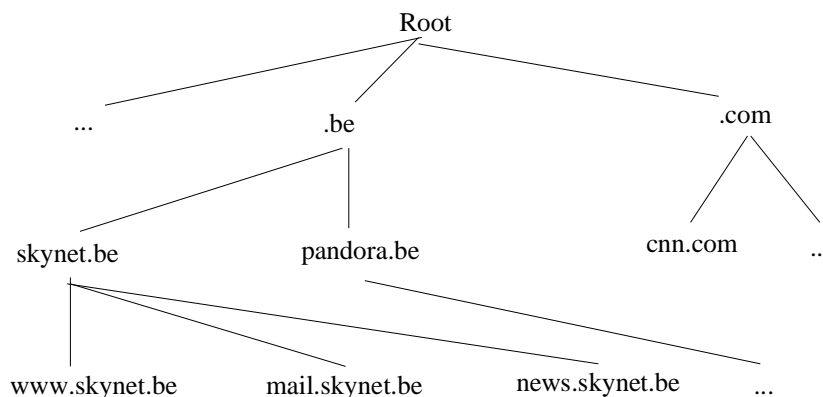
Je kan je de afgelegde weg van een NAT pakket als volgt voorstellen:



DNS

Omdat IP-adressen moeilijk te onthouden zijn is er een manier om aan een IP-adres een naam te geven. Een DNS is een database ergens op het netwerk/internet die bijhoudt welke naam overeenkomt met welk adres. De locale computer moet natuurlijk wel worden ingesteld om de DNS op te vragen.

DNS namen zijn doorgaan hiërarchisch gestructureerd.



IP in Linux

Linux is gebouwd op IP, daardoor is de ondersteuning van IP uitgebreider dan bij andere Operating Systemen.

In dit hoofdstuk is het de bedoeling om enkele Linuxprogramma's te overlopen die met IP te maken hebben.

ifconfig

Ifconfig kan zowel worden gebruikt om de stand van het netwerk te bekijken alsook het instellen van het netwerk. De meeste functies voor het wijzigen van de instellingen vindt men ook in andere programma's terug.

Indien men **ifconfig** uitvoert krijgt men alle actieve netwerkkaarten te zien.

```
eth0      Link encap:Ethernet  HWaddr 00:10:A4:B1:1C:5C
          inet addr:164.35.198.52  Bcast:164.35.198.255
          Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7253517 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5348 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0x1000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

lo is het locale netwerk en is altijd zichtbaar. De netwerkkaarten zijn genummerd met **ethx**. Dial-up verbindingen worden getoond als **pppx**.

Uniek aan een UNIX systeem is dat je de toestand kan zien van het aantal ontvangen/verzonden pakketten. Door naar het veld **errors** te kijken kan men ook een idee krijgen van de kwaliteit van het netwerk.

Met het commando "ifconfig -a" toon je alle netwerkadapters, ook al zijn ze inactief.

Als we nu het adres van eth0 willen wijzigen naar bijvoorbeeld 10.1.1.20 met netmask 255.255.0.0 geven we het volgende commando:

```
ifconfig eth0 10.1.1.20 netmask 255.255.0.0
```

Wil men de status van de netwerkadapter wijzigen geeft men het commando:

```
ifconfig eth0 up      (activeren)
ifconfig eth0 down    (deactiveren)
```

Bekijk **man ifconfig** voor verdere informatie.

dhcpcd

Dit is de DHCP client in de meeste Linux distributies (*pump* is een andere client, maar wordt minder frequent gebruikt). Om een DHCP adres te verkrijgen via **adapter eth0** geef je het commando:

```
dhcpcd eth0
```

netstat

Met netstat kan je onderzoeken welke poorten open staan of in gebruik zijn.

Indien je "netstat" intypt krijg je volgende uitvoer:

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 164.35.198.52:33117    10.201.:afs3-fileserver ESTABLISHED
tcp        0      0 164.35.198.52:33118    10.201.0.80:webcache    CLOSE_WAIT
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State           I-Node Path
unix   13     [ ]     DGRAM                    590    /dev/log
unix    3     [ ]     STREAM  CONNECTED      15016  /tmp/.X11-unix/X0
unix    3     [ ]     STREAM  CONNECTED      15015
unix    3     [ ]     STREAM  CONNECTED      14341  /tmp/.ICE-unix/1045
```

Unix domain sockets zijn niet zo belangrijk. Dit is vooral van belang bij communicatie tussen programma's.

Een tweede interessant gebruik van **netstat** is het onderzoeken van poorten die luisteren. Met "**netstat -an**" zien we volgende data.

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 0.0.0.0:515             0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:6000           0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:6001           0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:21             0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:23             0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:7100           0.0.0.0:*               LISTEN
tcp      0      0 164.35.198.52:33117    10.201.0.56:7000        ESTABLISHED
udp      0      0 0.0.0.0:32769          0.0.0.0:*

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags               Type           State         I-Node Path
```

Hieruit kunnen we afleiden dat onze computer luistert naar volgende poorten:

- 515 printersharing
- 6000 X:0
- 6001 X:1
- 21 ftp server
- 22 ssh server
- 23 telnet server
- 7100 font server

Met het commando "**netstat -M**" kunnen we bij een router, die als *Network Address Translator* dient, de actieve connecties bekijken die vertaald worden.

Een ander interessant commando is "**netstat -taunp**". Hiermee kunnen we zien welke programma's de poorten in gebruik nemen.

/etc/init.d/network

Met dit script kan je alle netwerkkaarten herstarten. Dit kan soms nodig zijn na een netwerkwijziging.

Je kan vier opties meegeven:

```
/etc/init.d/network start
```

Starten van alle netwerkkaarten

```
/etc/init.d/network stop
```

Stoppen van alle netwerkkaarten

```
/etc/init.d/network restart
```

Stoppen en starten van de netwerkkaarten

```
/etc/init.d/network reload
```

Gelijkaardig aan een "restart", maar bij een reload worden ook de netwerkdrivers herladen.

Cursus Linux

linuxconf

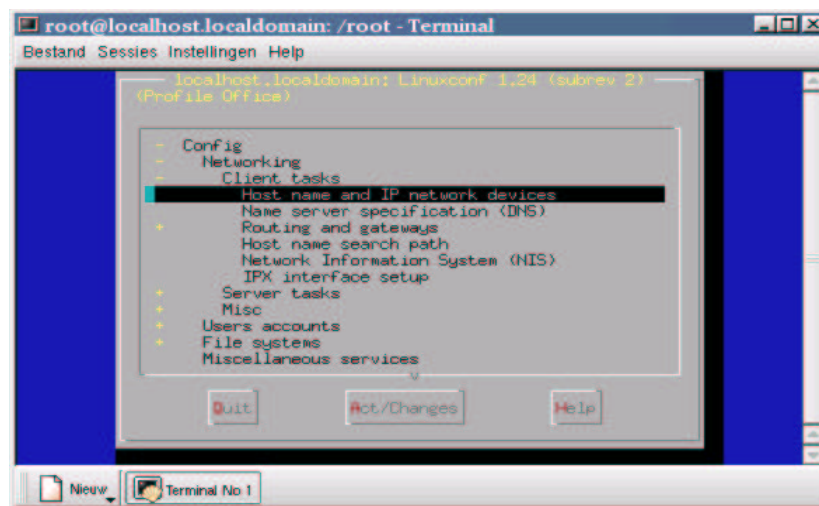
Linuxconf is een GUI configuratieprogramma. Je kan deze starten in de commandline (je krijgt dan een tekstversie) of in de grafische omgeving.

Met deze tool kan je zowat alles instellen wat er op een Linuxmachine in te stellen valt, waaronder ook de IP-configuratie.

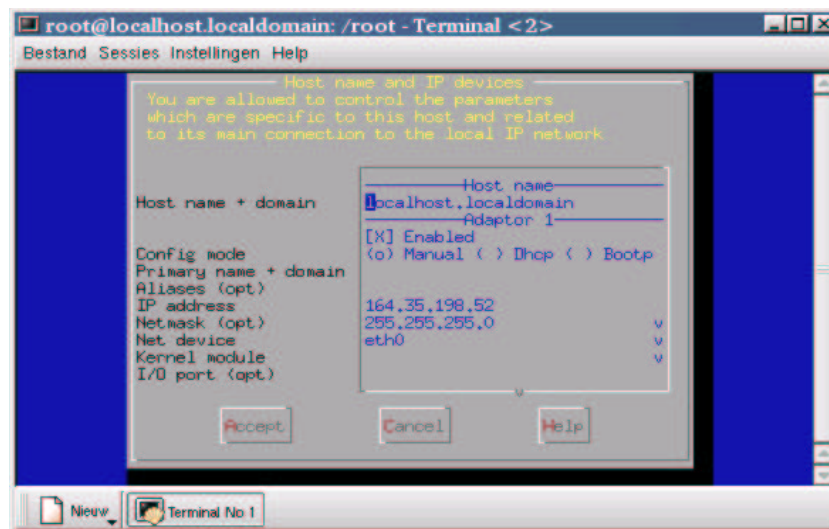
Om het IP-adres in te stellen in de tekstversie volg je volgende stappen:

(De TAB toets dien om van veld te veranderen. ENTER toets is selecteren/uitvoeren).

- Klap de submenu's uit tot volgend scherm ziet:

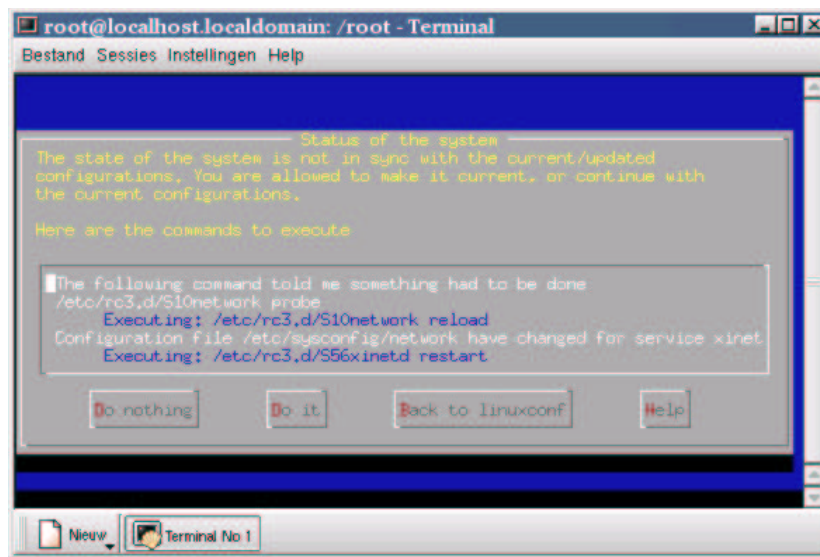


- Druk op ENTER om te selecteren

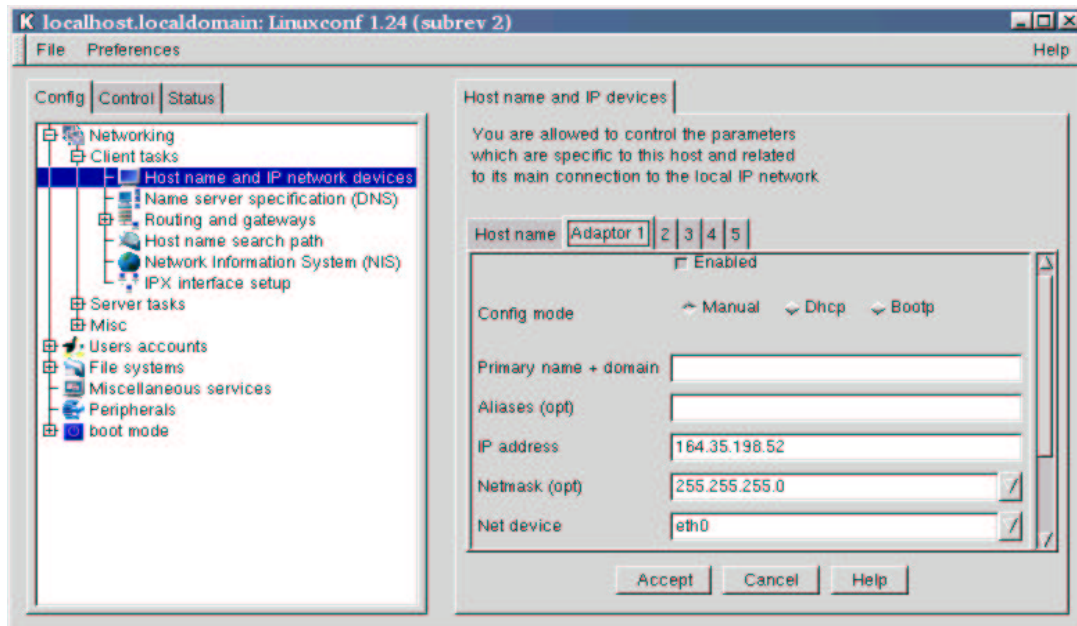


Cursus Linux

- Stel alles in naar wens. Velden met een "v" achter kan je uitklappen door te selecteren en dan CTRL-X te typen. Je krijgt dan een keuzelijst.
Indien je de instellingen hebt gemaakt mag je met TAB de *Accept* toets aanduiden en ENTER drukken om te selecteren.
- Gebruik *Quit* om Linuxconf af te sluiten. Bij de volgende vraag kies je *Do it*.



De Grafische versie werkt gelijkaardig, maar ziet er iets anders uit:



X-Window System

Het *X-window system* is een netwerk transparant GUI systeem. Op alle Unix systemen wordt X-Windows gebruikt. Het is ook mogelijk om X te gebruiken op een Windows PC (meestal met de bedoeling remote aan te loggen).

De *design* van het systeem lijkt ingewikkeld omdat het een netwerk gebruikt om gegevens op het scherm te tonen. Dit is volledig verschillend van het systeem in Windows dat niet gemaakt is om over netwerken te draaien.



Bij het gebruik wordt er op de PC een *X-server* gestart die luistert naar een bepaalde IP poort (poort 6000 voor display 0, 6001 voor display 1, enz.). Men start dan een GUI programma op. Dit programma onderzoekt de "shell variabele" **\$DISPLAY**. Het IP-adres dat daar is ingevuld wordt gebruikt om te connecteren met een X-server. Meestal draait die X-server op de locale machine, maar dit is geen vereiste.

Zo is het mogelijk een X-server te starten en in te loggen op een andere computer. Op die andere computer wijzig je de **\$DISPLAY** variabele:

```
export DISPLAY=192.168.1.5:0 (jouw IP-adres)
```

als je dan een GUI programma opstart op die andere PC dan zal je zien dat het venster verschijnt op jouw PC. Het programma draait wel degelijk op de andere computer.

Je mag niet vergeten op jouw locale PC enkele beveiligingen af te zetten. Dit doe je door het **xhost** commando.

xhost +	geef iedereen toegang
xhost <IP-adres>	geef een bepaald ip toegang
xhost -	weiger alle toegang

Starten van X

In de meeste gevallen ga je X-Windows enkel gebruiken om lokaal programma's te draaien.

Indien je in tekstmode opstart kan je X-Windows starten door het commando "startx". Indien je in grafische mode opstart wordt X-Windows als onmiddellijk opgestart met inlogscherf (respectievelijk **xdm**, **gdm** of **kdm**).

Om een tweede X server te starten gebruik je "startx -- :1".

Cursus Linux

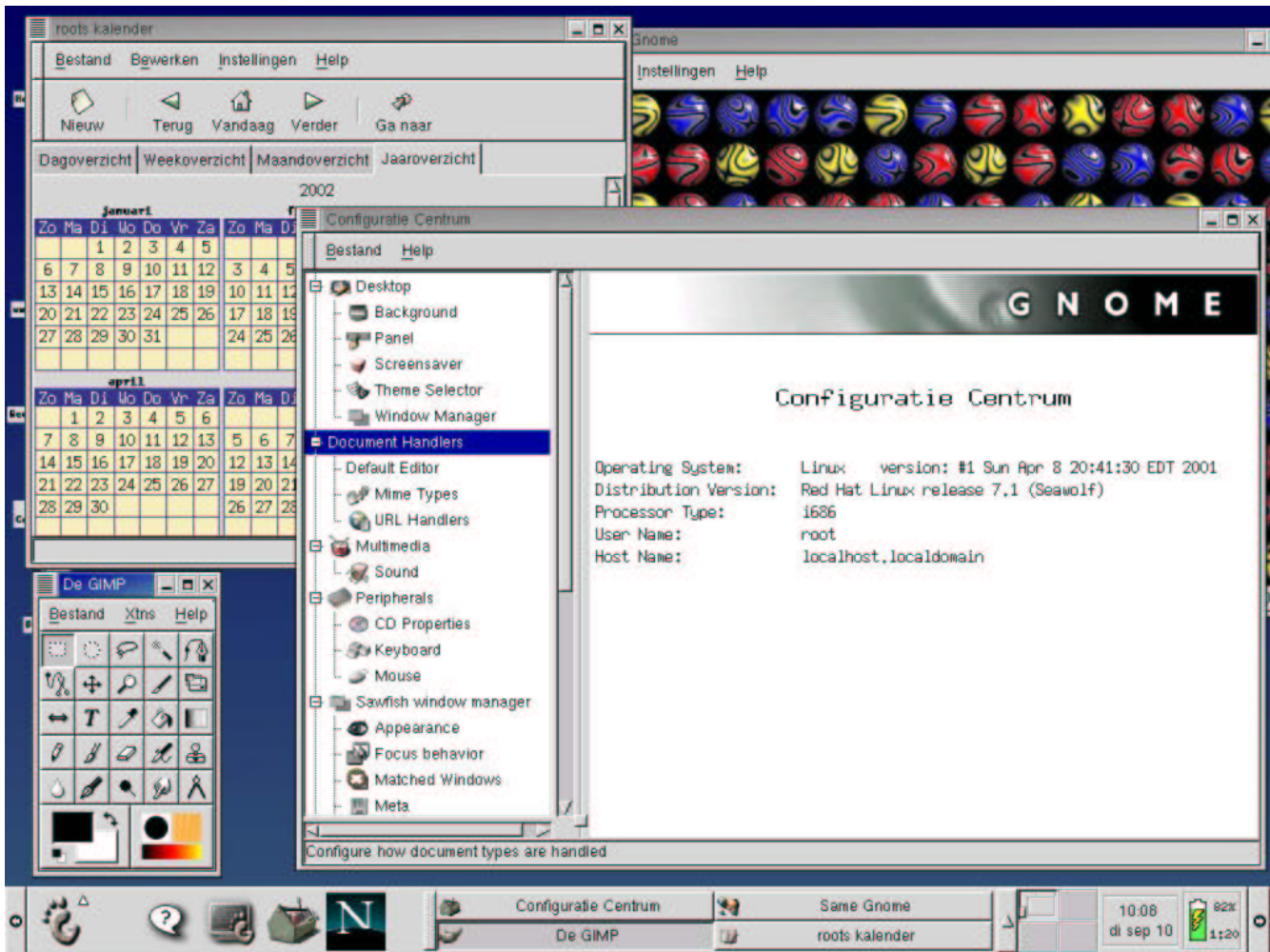
Window Managers

X is enkel het *framework*. Daarbovenop draaien windowmanagers die vensters en programma's beheren. Enkele voorbeelden van windowmanagers zijn **windowmaker**, **blackbox**, **fvwm**, **CDE**, **GNOME**, **KDE**, **Windowmaker**, **blackbox** en **fvwm** zijn *lightweight* windowmanagers en kunnen gebruikt worden op oude computers. In het verleden waren dit de standaard desktops op Linux systemen. **CDE** wordt vooral gebruikt om commerciële UNICES (Sun Solaris, HP-UX,...). **GNOME** en **KDE** zijn tegenwoordig dé desktops in Linux (en misschien binnenkort ook van andere UNICES). Deze twee dektops verschillen niet zoveel van elkaar in gebruik. Het enige grote verschil is dat KDE in *QT/C++* is geschreven terwijl GNOME gemaakt is in *GTK/C*. (QT en GTK zijn programmeerbibliotheken om GUI programma's te schrijven). Enkele screenshots:



KDE

Cursus Linux



GNOME

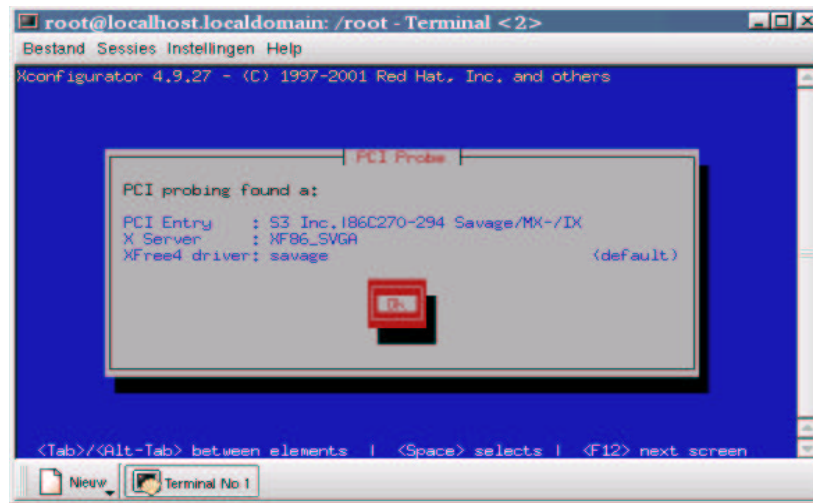
Het gebruik van beide windowmanagers lijkt een beetje op die van Windows. Het is wel aangeraden om een computersysteem te gebruiken dat krachtig genoeg is om deze windowmanagers te draaien.

instellingen

Bij de installatie kan je X instellen (inclusief de drivers). Indien er nadien iets moet gewijzigd worden kan dit via de tools:

Xconfigurator
XF86config

Je gebruikt best Xconfigurator omdat deze automatisch de videokaart kan detecteren. Bij sommige distributies zijn er GUI tools beschikbaar, zoals bijvoorbeeld **harddrake** in **Mandrake**.



Remote desktop

De klassieke manier (zie hierboven) om vanop afstand programma's te draaien is nogal omslachtig. Vaak is het ook gewenst om een volledig scherm inclusief windowmanager te draaien op een andere PC.

In *Windows* ken je waarschijnlijk programma's zoals **PCAnywhere** waarmee je de desktop kan overnemen. In *Linux/Unix* neem je niet echt een dektop over: iedere gebruiker logt in op een eigen desktop. De gebruiker die aan de PC zelf zit heeft dus geen last als andere mensen aanloggen.

XDMCP gaan we in *Unix/Linux* gebruiken. Dit is een protocol in het X-Window System die ervoor zorgt dat we vanop afstand inloggen.

Om dit in te stellen moet je volgende zaken configureren (er wordt vanuit gegaan dat **gdm** wordt gebruikt):

- In `/etc/X11/gdm/gdm.conf`
Bij sectie `[xdmcp]` moet de optie als volgt staan:
`Enable=1`
- In `/etc/X11/xdm/Xaccess`
Specifieer de IP-adressen van de hosts die mogen aanloggen.
Indien je iedereen wil toelaten zet je ergens in het bestand een `"*"`.

Bij een herstart van **gdm** zal je nu vanop afstand kunnen inloggen

Bij gebruik van **xdm** of **kdm** gebruik je andere bestanden. Kijk hiervoor naar:

`/etc/X11/xdm/xdm-config`

Nieuwere versies van **kdm** gebruiken `/usr/share/config/kdm/kdmrc`

Je kan ook nog de fontserver herinstellen zodat de andere X-server jouw fonts kan gebruiken. De instelling is op bijna iedere Linux distributie anders. Om deze reden geen we er ook niet dieper op in. Indien je twee gelijkaardige linuxsystemen gebruikt is dit zowiezo niet nodig omdat op beide systemen gelijkaardige fonts worden gebruikt. Indien er niet dezelfde fonts voorhanden zijn kunnen de letters er iets anders uitzien.

Je kan op **XDMCP** ook inloggen vanaf een Windows PC. Hiervoor moet je een X-server downloaden (tip: www.tucows.com) zoals WinaXe, X-Win32, enz... De meeste versies zijn shareware.

Indien je inlogt vanop een Windows PC kan je meestal toch geen gebruik maken van een fontserver, en moet je de moeite niet doen om die in te stellen.

Vanop een Linux PC kan je aanloggen met het volgende commando:

```
X -query <IP-adress remote>
```

Om de desktop te tonen op een tweede DISPLAY geef je het commando:

```
X :1 -query <IP-adress remote> (:1 is het nummer van de display)
```

Indien je gebruik wil maken van een fontserver ziet je commando er als volgt uit:

```
X -query <IP-adress remote> -fp tcp/10.201.0.56:7100
```

```
X :1 -query <IP-adress remote> -fp tcp/10.201.0.56:7100
```

Nadelen aan XDMCP is dat je gebruikt maakt van meerdere poorten tegelijkertijd. Dit protocol is dus niet bruikbaar om te *routen* en dus onmogelijk om te gebruiken over het internet. Hiervoor kun je VNC gebruiken.

VNC

VNC staat voor *Virtual Network Computing*. Het is een *gratis* alternatief voor producten zoals PCAnywhere. VNC draait zowel op Windows, Mac en Linux/Unix (zowel server en viewer). In Linux heeft VNC ongeveer dezelfde werking als X (iedere gebruiker die inlogt krijgt een eigen desktop). De Windowsversie werkt dan weer gelijkaardig aan PCAnywhere (de gebruiker neemt de desktop over). In Windows kan er dus niet meer dan één gebruiker aanloggen (VNC heeft in Windows wel de optie om met meerdere op één scherm aan te loggen, maar slechts één gebruiker kan de muis/toetsenbord gebruiken).

VNC gebruikt één enkele poort en is geschikt om over internet en netwerken met lage bandbreedte te werken (gebruikte server poort is TCP 5900).

In de meeste Linuxdistributies is er een server en viewer inbegrepen (indien niet geïnstalleerd staat deze meestal op de CD).

Ga ook naar deze website om meer te weten over VNC:

<http://www.uk.research.att.com/vnc/>

Scripts



Met scripts kunnen we kleine programma's schrijven.

In UNIX hangt de taal waarin je de scripts schrijft af van welke shell je gebruikt. De standaard Linux shell is **bash**, maar **cs**h (C-shell) is ook voorhanden. In dit hoofdstuk bespreken we hoe je een programma kan schrijven met de *bash-shell*.

Andere veel gebruikte scripting talen die geen shell gebruiken zijn *perl* en *python*.

Om het systeem aan te geven dat we de *Bash-shell* gebruiken zetten we in het begin van het programma: **#!/bin/bash**

Net zoals andere programmeertalen heeft de *bash-shell* ondersteuning van selecties en iteraties.

We kunnen aan een programma ook argumenten meegeven. Je moet hiervoor het scripts starten met bijkomende data achter het commando. Die data wordt dan geïnterpreteerd als argument(en).

De bedoeling van volgende voorbeeldscripts is om op een praktische manier de syntax van *Bash-shell* te bekijken.

Maak een nieuwe map aan in de home-dir waarin je de ingetypte scripts bewaart. Een script start je door de naam in te typen voorafgegaan van een `./` (*./script*).

Voor dat je een script uitvoert moet je jezelf ook nog rechten geven om dit bestand uit te voeren. Doe dit met het commando `chmod 755 scriptnaam`.

Eenvoudig script

We beginnen met een script dat enkele lijnen op het scherm afprint.

```
#!/bin/bash
echo Welcome to the world of script files
echo -n "The current time and date are: "
date
```

- Waarvoor dient de optie `-n` in de derde lijn?
- Waarom gebruiken we daar `""` tekens?

Argumenten

Je gaat nu een script maken dat gebruik maakt van argumenten. Je kan argumenten meegeven door deze achter de bestandsnaam te plaatsen bij het starten van een script.

Met "\$#" vraag je het aantal argumenten op.

```
#!/bin/bash
echo There are $# arguments
```

Wat is de uitvoer volgende twee commando's? (wijzig scriptnaam indien nodig)

```
./script2 een twee drie vier
./script2 "een twee drie vier"
```

Selectie

```
#!/bin/bash
if [ $# = 0 ]; then
    echo There are no arguments
else
    echo There are $# arguments
fi
```

De selectie heeft de structuur "if [_conditie(s)]; then (doe iets) else (doe iets) fi". De "else" tak mag weggelaten worden. De "fi" toont het einde van dit blok. Logische operatoren (NOT, OR, AND) worden voorgesteld als !, || en &&.

Voer volgende lijnen bij achteraan het script.

```
if [ -e $1 ]; then
    echo The file: $1 exists.
fi
if [ ! -e $1 ]; then
    echo The file: $1 does not exist.
fi
```

Met de optie "-e" onderzoek je het bestaan van een bestand (zie **man bash** voor meer info). De "\$1" staat voor de variabele dat het eerste argument bevat.

Voer het programma uit met enkele testwaarden. Probeer het programma ook eens uit te voeren **zonder** argumenten.

Wat gaat er mis? Pas het script aan zodat het correct werkt.

For-iteratie

Met een "for"-loop kunnen we een iteratie uitvoeren waarvan we vooraf weten hoeveel keer we het moeten uitvoeren.

```
#!/bin/bash
for i in 1 2 3 4 5 6 7 8 9 10;
do
    echo $i
done
```

Wijzig lijn 2 in:

```
for i in `seq 1 10`;
```

Je zal merken dat dit dezelfde uitwerking heeft.

While-iteratie

Indien we voorafgaand niet weten hoeveel keer we door een lus moeten, gebruiken we best een "while".

Het volgende script zoekt een woord op zolang er invoer is, anders stopt het.

```
#!/bin/bash
word="anything"
set word
while [ "$word" != "" ]; do
    echo -n "Enter a word to check (Return to exit): "
    read word
    if [ "$word" != "" ]; then
        grep $word /usr/share/dict/words
    fi
done
```

Waarom zetten we in de *if-test* "\$word" tussen aanhalingstekens?

Vergelijkingen

Probeer van het volgende programma te achterhalen wat het precies doet.

- Bij invoer van welke getallen krijgen we welke uitvoer?
- Wat betekenen `-le` `-lt` `-ge` `-gt` `-eq` `-ne`? (`-eq` en `-ne` komen niet voor in het programma)

Gebruik de handleiding/man pages om dit op te zoeken.

```
#!/bin/bash
declare -i number
class=3
if [ $# == 0 ]; then
    echo No number to classify
else
    number=$1
    if [ $number -le -100 ]; then
        class=0
    fi
    if [ $number -ge 0 ] && [ $number -le 100 ]; then
        class=1
    fi
    if [ $number -gt 1000 ] && [ $number -lt 2000 ]; then
        class=2
    fi
    echo The number is in class $class
fi
```

Met "declare" kan men een variabele declareren. Het declareren is niet verplicht, maar kan nuttig zijn indien je de variabele wil blokkeren voor andere types van data. De optie "-i" forceert een *integer*. (zie **man pages** voor de syntax van andere declaraties)

Case

```
#!/bin/bash
echo; echo "Hit a key, then hit return."
read Keypress
case "$Keypress" in
    [a-z] ) echo "Lowercase letter";;
    [A-Z] ) echo "Uppercase letter";;
    [0-9] ) echo "Digit";;
    *      ) echo "Punctuation, whitespace, or other";;
esac
```

Net zoals in andere programmeertalen heb je ook "case" ter beschikking. Bestudeer de syntax van **case** (schrijf deze eventueel op).

Internet

Linux omvat een aantal toepassingen om het internet te betreden. De lijst van internet programma's is enorm omvangrijk, maar we gaan de populairste *tools* even bekijken.

Internetverbinding

Om op het internet te geraken kan je verschillende soorten verbindingen hebben. De meest gebruikte zijn *dial-up*, *kabel* en *ADSL*. Al deze verbindingen worden zonder enige problemen door linux ondersteunt .

Dial-up

Voor een *dial-up* heb je een modem nodig. Alle **externe seriële** modems zijn ondersteund zonder enig probleem, alsook interne modems met een **seriële** connectie (meestal zijn dit modems met een *ISA* interface). Men moet opletten met *winmodems*, dit zijn goedkope **interne** modems met *PCI*-bus. De modems worden volledig softwarematig bestuurd door een *driver*. In de meeste gevallen heeft de fabrikant geen driver geschreven voor Linux.

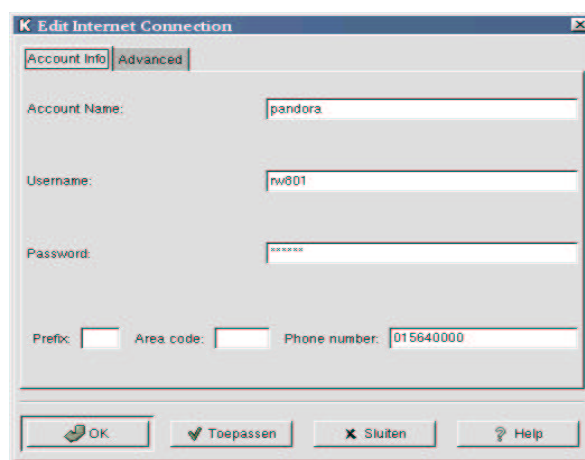
Er zijn niettegenstaande enkele *drivers* beschikbaar op <http://www.linmodems.org>.

De configuratie van de *dial-up* is verschillend bij iedere distributie omdat ze ieder een andere tool meeleveren. Bij RedHat is dit bijvoorbeeld "**Dial-Up Configurator**". Al deze tools werken met een *wizard* invoer zodat de configuratie niet moeilijker is dan ze moet zijn.

Voor diegenen die een *dial-up* verbinding willen starten vanuit de **console** kunnen de handleiding bekijken van *wvdial* (en zijn configuratiebestand */etc/wvdial.conf*).

Kabel internet

Zowel Telenet als andere kabelproviders in Europa leveren een externe modem met netwerkaansluiting. De modem wordt dus aangesloten op een netwerkkaart. Het is voldoende om deze netwerkkaart in te stellen zodat deze een **DHCP-adres** krijgt. Je mag niet vergeten om in de browsers een proxyserver in te vullen (bij Telenet is dit **proxy.pandora.be:8080**). Indien men van Telenet een Silicom USB adapter heeft ontvangen kan men die vervangen door een normale ethernetkaart, of als men avontuurlijk is <http://bigblue.dyndns.org:8080/usb>. Gebruikers van Mandrake 8.1 of hoger zullen merken dat hun adapter zonder problemen werkt.



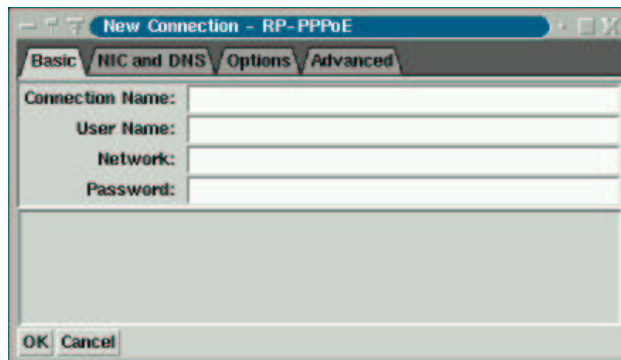
ADSL

In België gebruiken we *PPPoE* als protocol (net zoals in Frankrijk, USA,...). Op Linux kan je alle **externe ADSL-modems** met *netwerkaansluiting* gebruiken zonder enige problemen. Het is afgeraden om een interne of USB modem te kopen.

Om verbinding te maken met het ADSL-netwerk moet je een *PPPoE*-client installeren. In de meeste distributies zit deze

er al in, indien dit niet het geval is bezoek dan <http://www.roaringpenguin.com/pppoe>. De setup gebeurt via tekstbestanden. Instellingen worden geschreven in */etc/ppp/pppoe.conf*. Voor de beginnende gebruiker zijn er ook grafische setup-tools voorzien. Eén voorbeeld is **tkpppoe** waarvan je een screenshot kunt bekijken. Iedere distro levert wel één of andere setuptool voor ADSL.

In Nederland wordt er het *PPTP*-protocol gebruikt. Ondersteuning hiervoor is beschikbaar op <http://www.adsl4linux.nl>.



Browsers

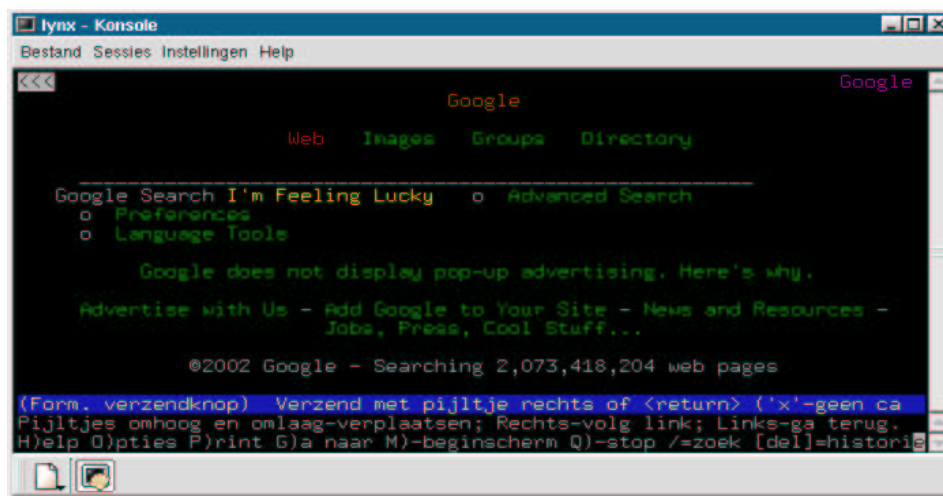
Het allereerste waar men aan denkt bij het woord *internet* is *surfen*.

Lynx

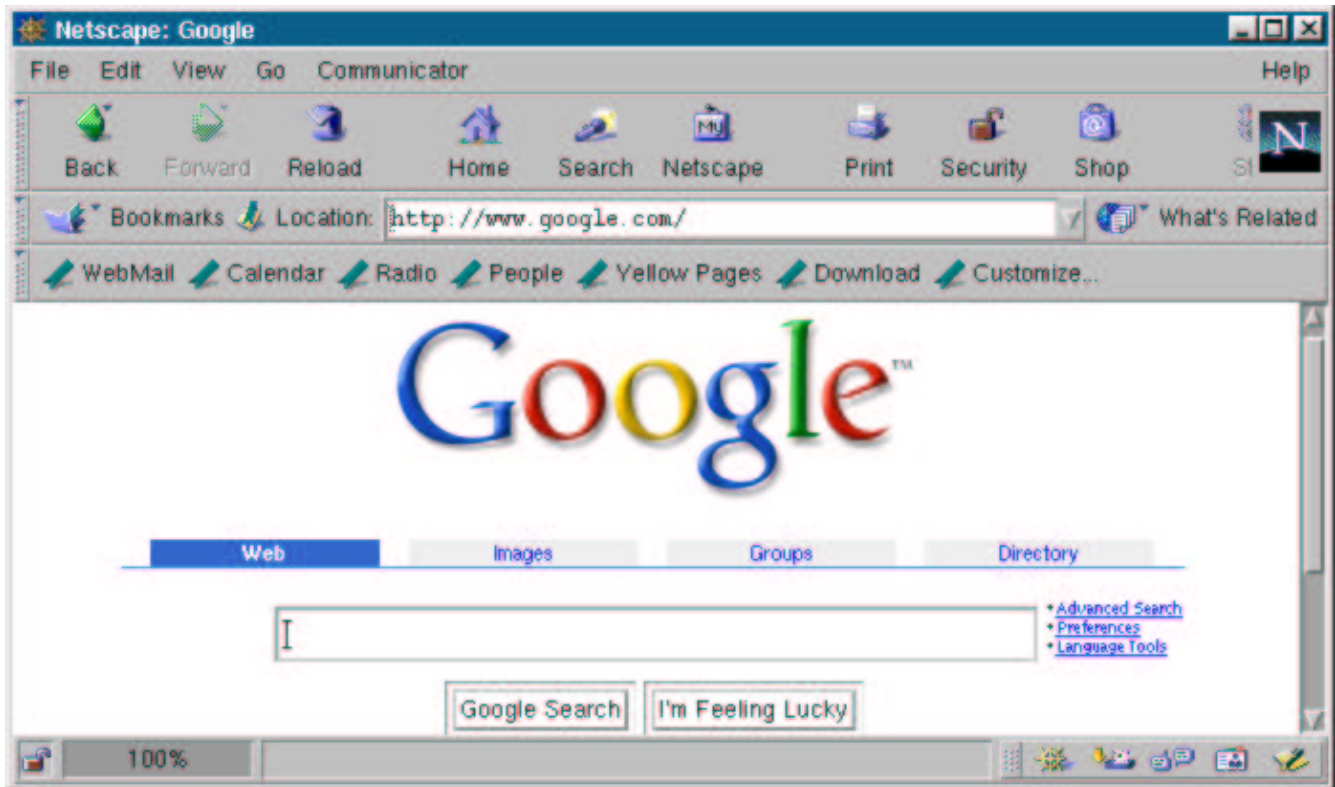
Indien men in de *console* wil surfen kan men **lynx** gebruiken (zie screenshot).

Lynx is een vrij eenvoudige textbrowser met ondersteuning voor *http* en *ftp*.

Minpunten: geen ondersteuning voor grafische gegevens (suprise :-)) en frames.



Netscape 4.x



Netscape 4.x is al geruime tijd een vaste waarde in Linux. Zowat iedere distributie bevat deze versie van Netscape.

Netscape heeft een heel goede ondersteuning voor standaarden, maar jammer genoeg is deze versie van Netscape een beetje "outdated".

De stabiliteit van deze browser is ook niet om over naar huis te schrijven (vooral in combinatie met JAVA doet deze browser soms raar).

Netscape 4.x is een browser die enkel in **binaire** vorm te verkrijgen is.

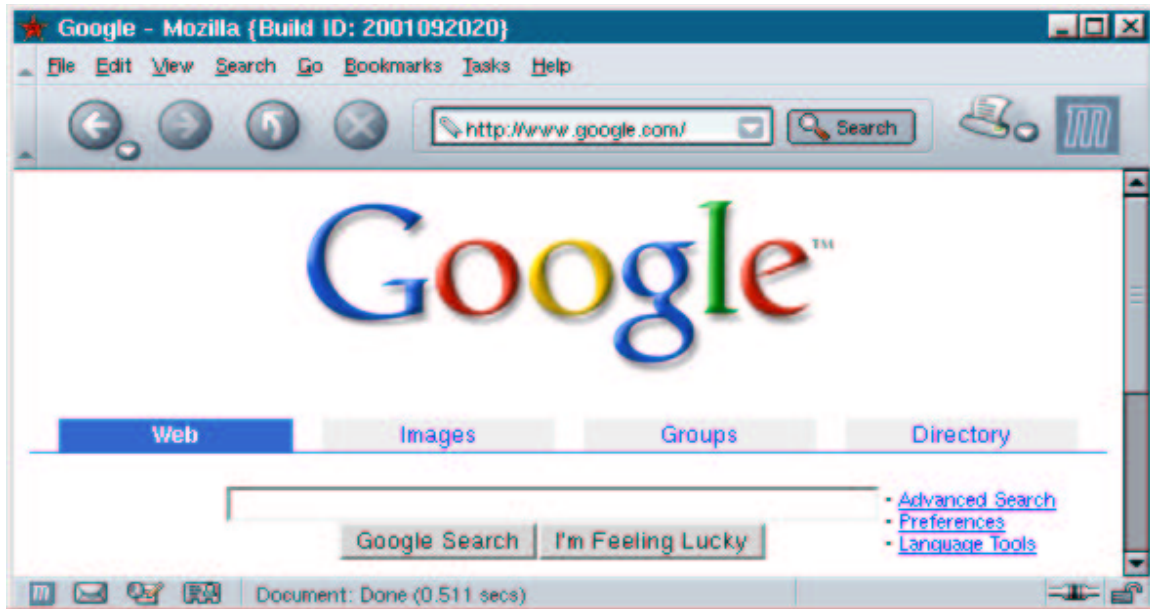
Mozilla / Netscape 6.x

Gelijklopend met de *Windows*versie wordt er gewerkt aan een nieuwe Netscape. Dit project heeft **Mozilla** (<http://www.mozilla.org>). **Mozilla** is een open-source project, dus men kan de volledige broncode downloaden en het programma aanpassen/compilieren.

Netscape 6.x is eigenlijk **Mozilla** + JAVA + plugins (flash,...) + extra programma's.

Op tragere computers laadt deze browser vrij traag, maar op een voldoende snelle machine is het comfortabel surfen.

Cursus Linux



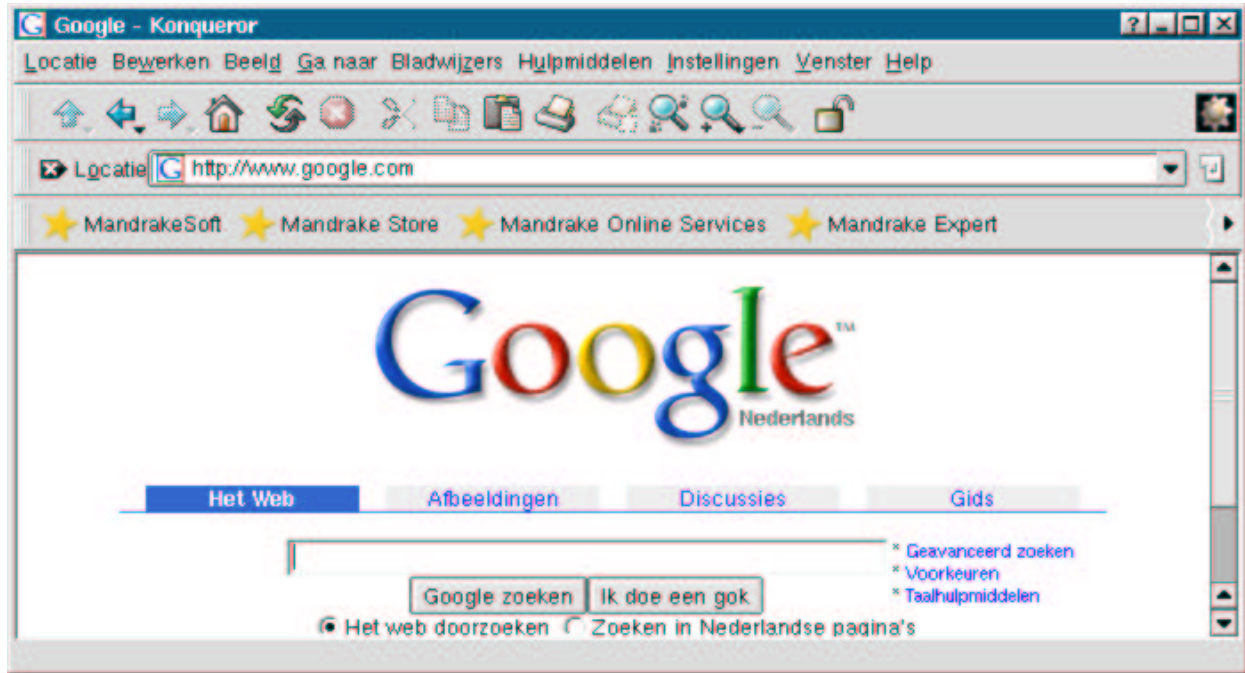
Galeon

Galeon is een project dat is gegroeid uit GNOME. De browser heeft de HTML-engine van **Mozilla**, maar met een interface herschreven in GTK (de GUI Toolkit van GNOME). De snelheid van deze browser ligt veel hoger dan **Mozilla**.



Konquerer

De programmeurs van *KDE* hebben op een zeer korte tijd een volledige browser geschreven. Zelfs de *HTML-renderer* hebben ze zelf geschreven. De browser is enorm snel en werkt heel comfortabel.



Opera

Gelijkaardig aan de *Windows*versie. Samen te vatten in drie woorden: Snel, betrouwbaar en stabiel. Bezoek ook <http://www.opera.com>.

Internet Mail

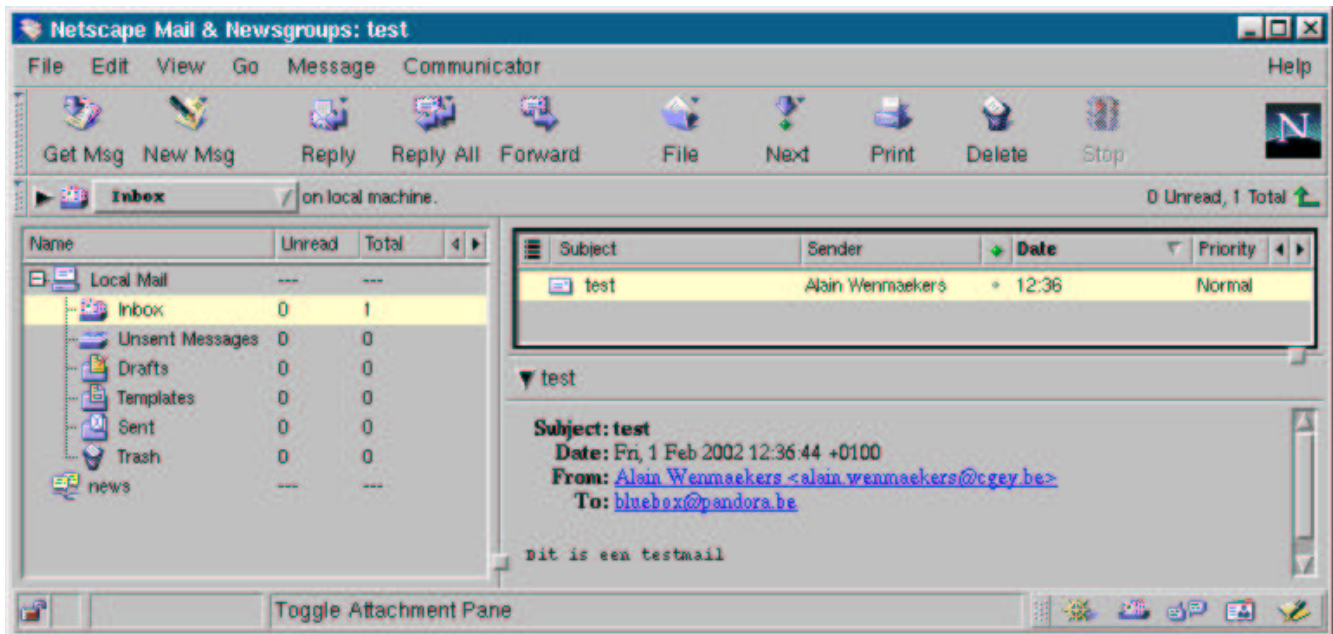
Net zoals het surfen is het versturen en ontvangen van e-mail een onmisbare applicatie op het wereldwijde internet. De keuze tussen e-mailapplicaties is dan ook enorm.

De e-mail applicaties voor de *console* werd al eens eerder besproken in een vorig hoofdstuk. Je zal merken dat de populairste *e-mail clients* van dezelfde makers komen als de browsers.

Netscape Messenger 4.x

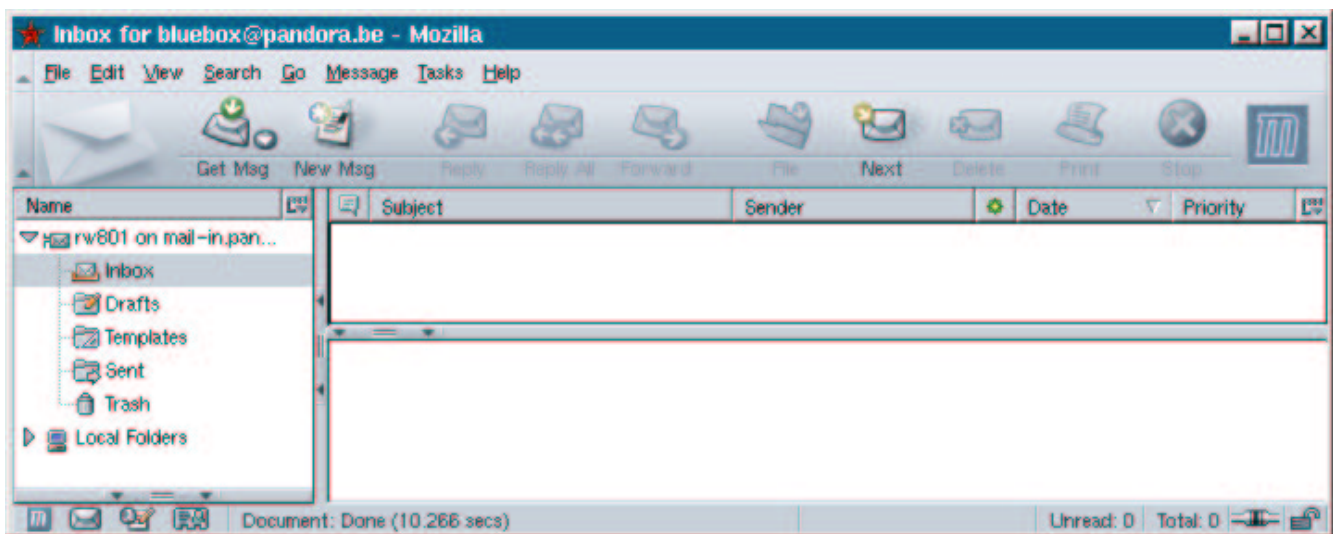
Bij Netscape 4.x zit er een e-mail client. Deze heeft de naam **Messenger** meegekregen. Messenger ondersteunt alles wat men in een e-mail client mag verwachten, behalve dat deze versie ook al wat tekenen van ouderdom geeft.

Cursus Linux



Mozilla / Netscape 6.x Messenger

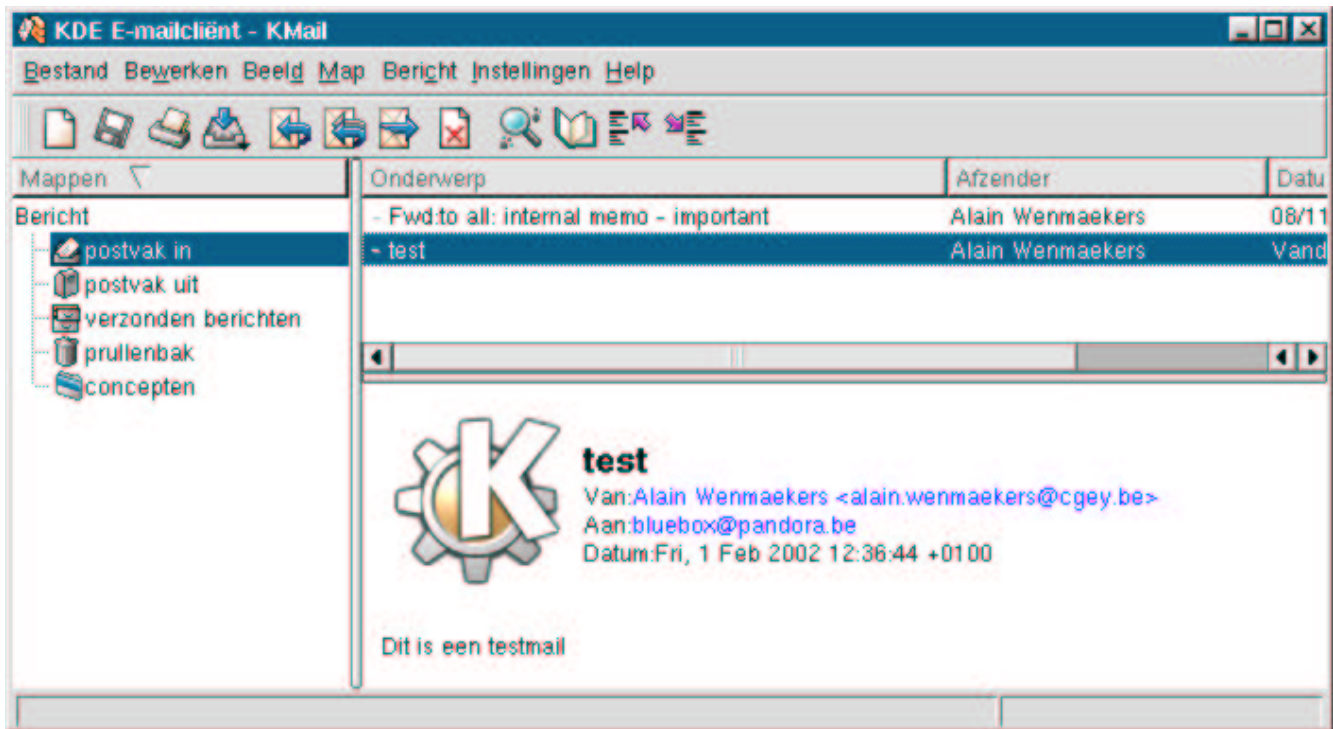
Net zoals bij de browsers is er een nieuwere versie in ontwikkeling. De werking is gelijkaardig, maar er is nog een uitgebreidere ondersteuning van protocols en de *User Interface* heeft een face-lift ondergaan.



KMail

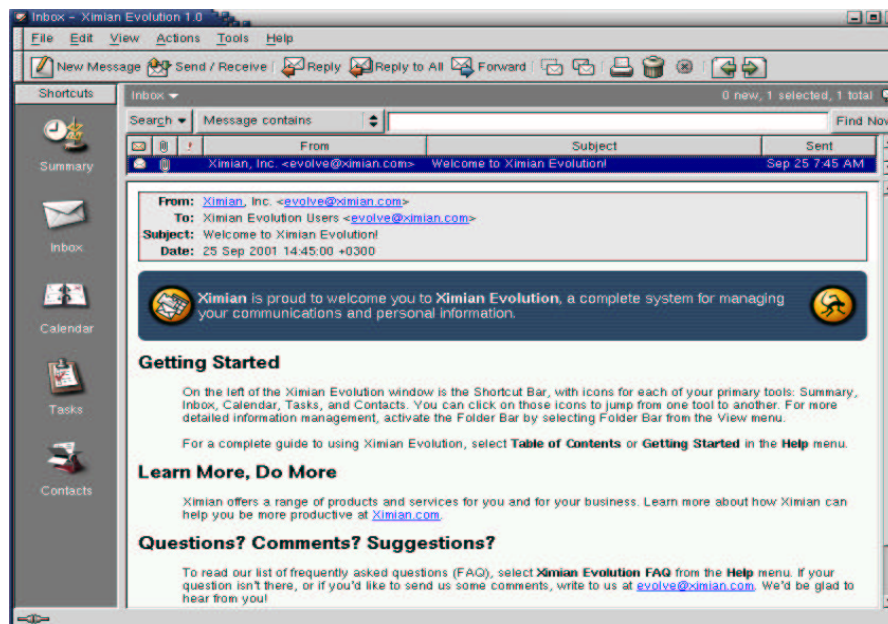
KMail is de e-mail client van KDE. Doorwinterde e-mail gebruikers zullen al onmiddellijk merken dat deze applicatie een aantal geavanceerde *features* mist, maar toch is deze heel bruikbaar.

Cursus Linux



Evolution

Evolution is de GNOME-mail client. Dit programma is als laatste van de opgenoemde mail clients uitgekomen. Evolution steekt *Microsoft Outlook* met gemak voorbij als het gaat om *features* en *collaboratietools* (adresboek, scheduling,...). Indien jouw distributie is uitgerust met deze client dan is dit zeker een aanrader.



File Transfer Protocol

FTP-servers worden vaak gebruikt om bestanden te verdelen. De meest voorkomende manier om iets te downloaden van een FTP-server is door gebruik te maken van een webbrowser. Alle webbrowsers ondersteunen FTP om te *downloaden*.

Als je iets wil *uploaden* ben je verplicht om gebruik te maken van een FTP-client.

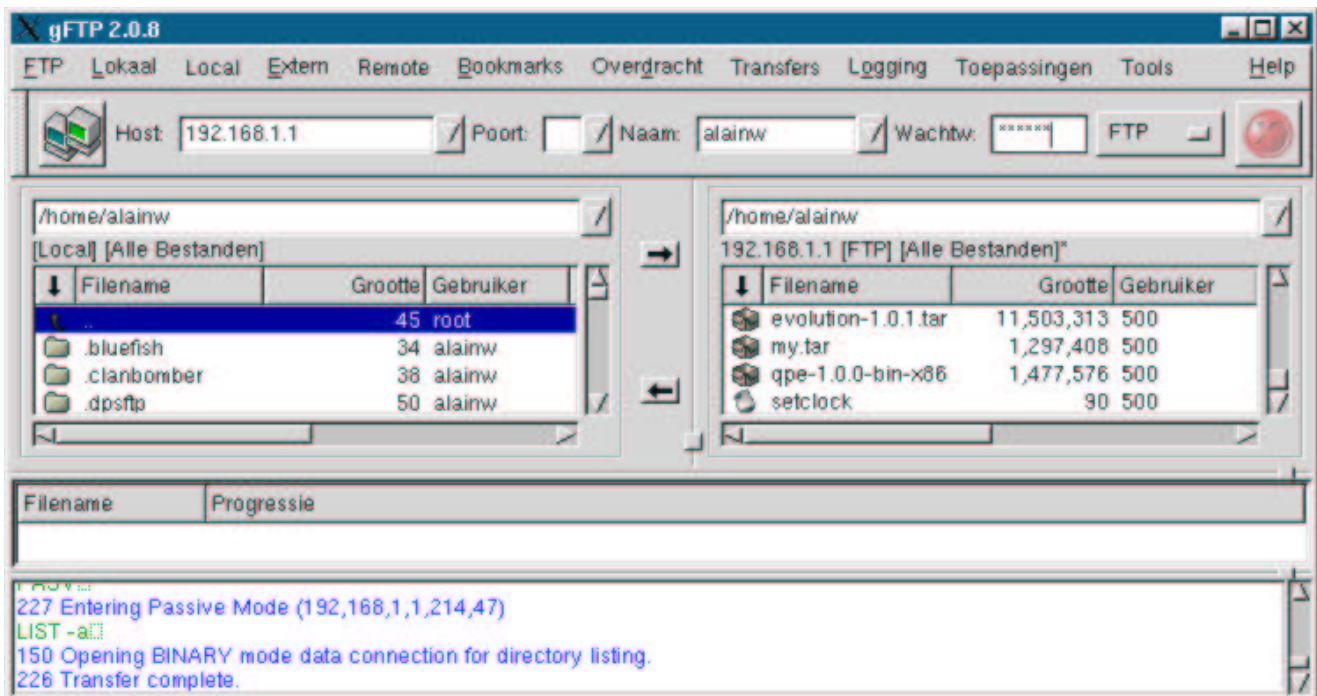
Onder de *console* heb je bijvoorbeeld het **ftp** programma.

De *commandline syntax* is niet echt moeilijk.

ftp ftp.anyserver.com

Je typt dan je gebruikersnaam en paswoord wanneer er om gevraagd wordt. Indien je geen paswoord hebt, gebruik dan **ftp** (voor gebruiker) en jouw e-mail adres als paswoord. Bestanden downloaden gaat via het `get` commando (of `mget` als je de bestanden wil downloaden met een wildcard). Om bestanden te uploaden gebruik je `put` (of `mput`). Navigeren op de server kan met de `cd` en `ls` commando's (dezelfde commando's als in de console). Als je wil uitloggen typ je `bye`.

Als je een GUI ter beschikking hebt kan je die beter gebruiken omdat dit gebruiksvriendelijker is. De meeste distro's leveren een aantal clients, maar de meeste gekende is **gftp**. De werking is ongeveer gelijkaardig met FTP-clients uit *Windows*.



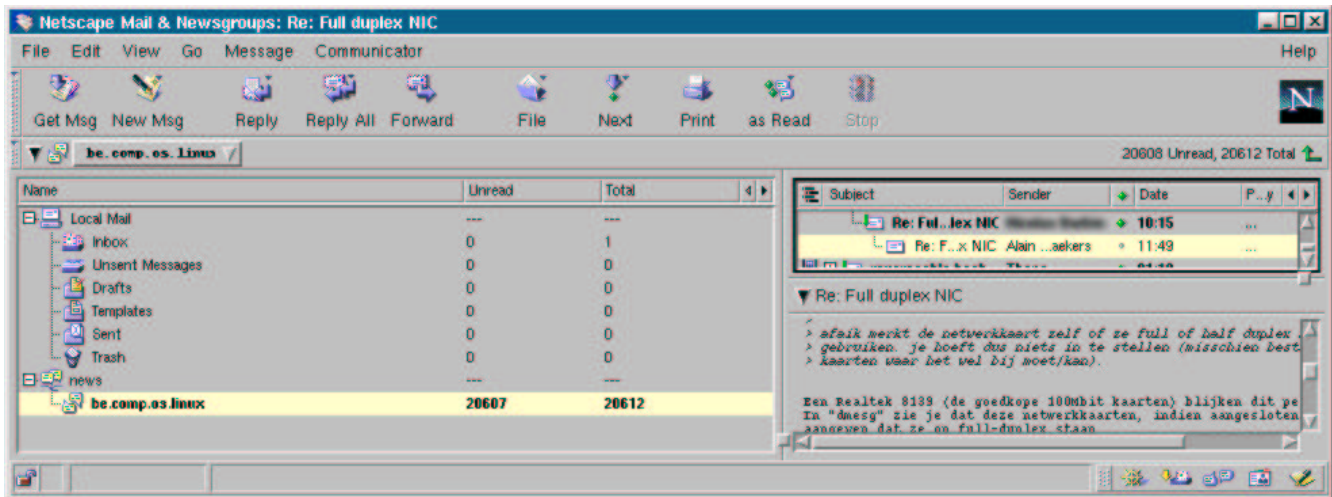
Cursus Linux

Nieuwsgroepen

Nieuwsgroep-applicaties zijn er ook genoeg. We beschrijven even de meest gekende.

Netscape 4.x Collabra

Netscape heeft een eigen nieuws-client. De stabiliteit is niet 100%, maar degelijk genoeg om ermee te werken.

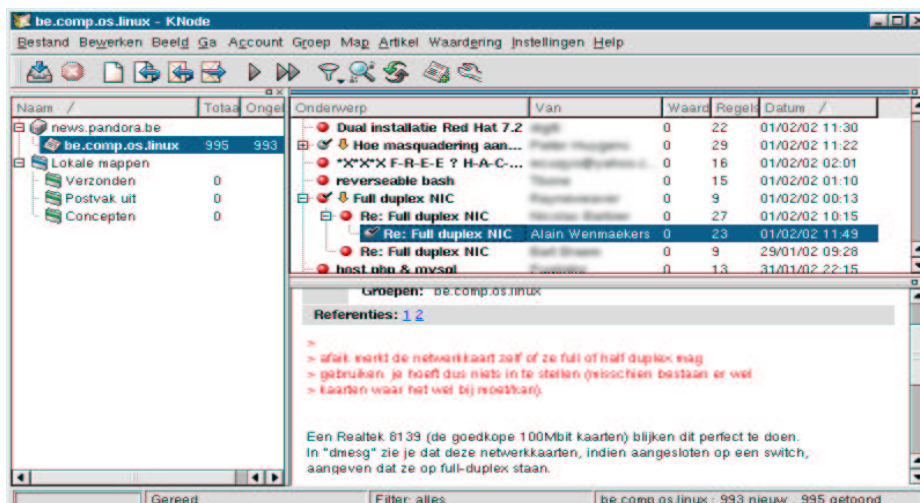


Mozilla / Netscape 6.x

Mozilla is volledig gelijkaardig aan Netscape 4.x (behalve de ge-update *user interface*). De stabiliteit is enorm verbeterd in deze versie.

KNode

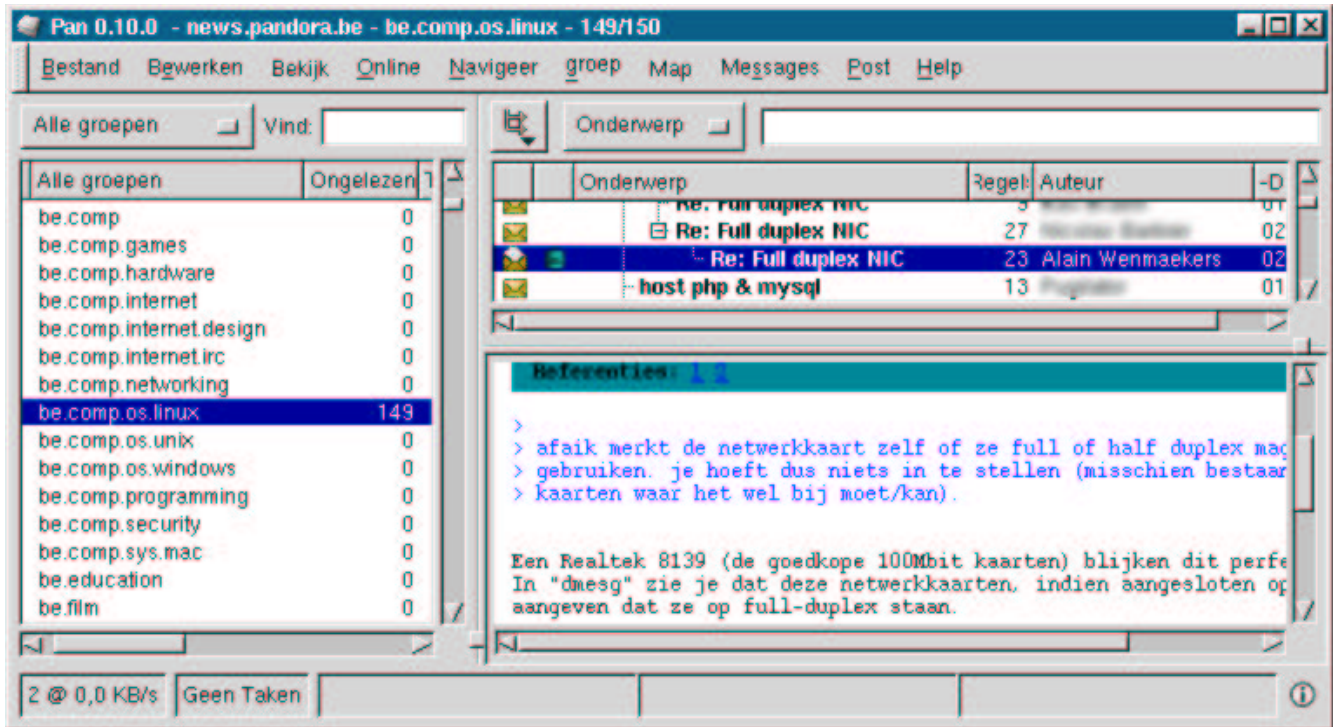
Net zoals KMail en Konquerer is dit een product van de KDE-programmeurs. De gebruikersinterface is simpel, snel en intuïtief.



Cursus Linux

Pan

Dit is de nieuws-client van GNOME. De gebruikersinterface is soms wat minder intuïtief, maar deze client ondersteunt zowat alles wat de geavanceerde gebruiker verlangt.

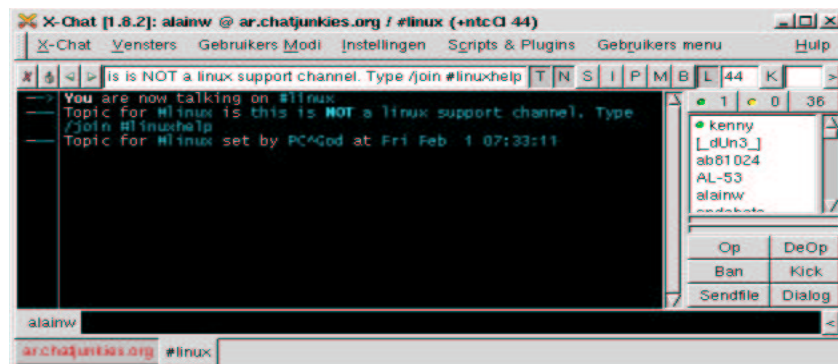


Chat

Ook een Linux-gebruiker kan gaan beginnen chatten met heel de wereld. Er zijn verschillende kanalen voorhanden zoals IRC, ICQ, MSN Messenger, AOL,...

X-Chat

Om te chatten op IRC zijn er genoeg clients voorhanden. Eén van de meest gebruikte versies op Linux is **X-Chat**.

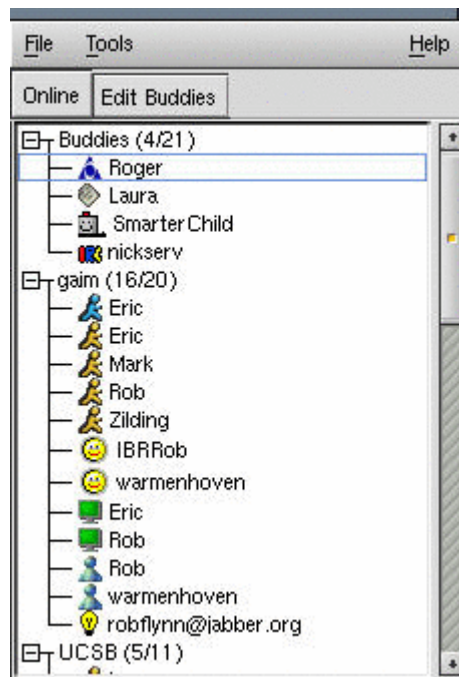


Licq

De gebruikers van ICQ kunnen in Linux ook hun favoriete chat-service gebruiken. Er zijn geen speciale *features* zoals SMS en voice-chat beschikbaar, maar de gewone dingen zoals messages, chat en file transfer werken perfect.

Gaim

Gaim is ontstaan als een clone voor AOL-messenger, maar ondertussen ondersteunt deze client zowat alle chat-servers die men kan inbeelden (vb. MSN, ICQ, AOL,...).



Office

Linux heeft net zoals **Windows** een hele reeks programma's om op kantoor te gebruiken. **Microsoft Office**, hét belangrijkste officeprogramma in **Windows**, heeft sinds kort enkele concurrenten.

De bedoeling van dit deel is om de office-omgevingen in Linux in detail te bekijken. Er zijn geen oefeningen verbonden aan dit hoofdstuk, maar na deze les mag wat "gespeeld" worden met de programma's die beschreven zijn.

Koffice

KDE heeft een eigen *Office-suit*. De ontwikkelaars van KDE staan er altijd voor garant om programma's te maken die gemakkelijk zijn in gebruik. De eerste versies van **KOffice** waren nogal onstabiel en hadden te weinig "features", maar op zeer korte tijd is deze Office van een speelgoedje naar een bruikbare *tool* uitgegroeid. Sinds de allerlaatste versie is het zelfs mogelijk om **Microsoft Office** bestanden in te lezen.

KOffice heeft net als KDE de reputatie héél snel te werken als je over degelijke hardware beschikt.

KOffice bestaat uit verschillende onderdelen:

- KWord (tekstverwerker)
- KSpread (spreadsheets)
- KPresenter (presentaties)
- KChart (grafieken)
- Kontour, vroeger *KIllustrator* (tekenprogramma)

We gaan enkele onderdelen eens van naderbij bekijken.

1. KWord

KWord is een simpele en effectieve tekstprocessor. Het heeft minder toeters en bellen dan **Microsoft Word**, maar voor de meeste zaken voldoet deze ruimschoots.

Het enige wat er ontbreekt is een degelijke *spellingchecker*. In de toekomst zal er wel een Engelse spellingscontrole worden ingebouwd, maar dan nog ga je een tijdje moeten wachten op een Nederlandstalige versie.

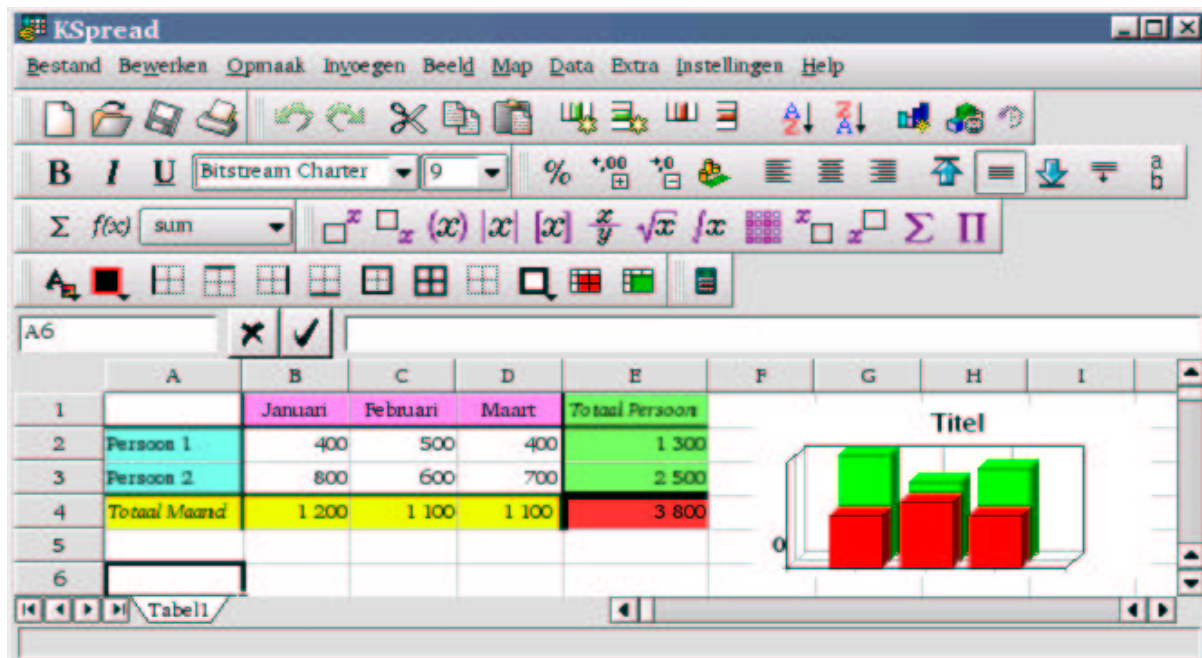
KWord is een onderdeel van KDE en beschikt over een Nederlandstalige plugin.

Cursus Linux



2. KSpread

Net zoals KWord is dit een heel eenvoudig *spreadsheet programma*. Er is een ondersteuning voor grafieken. De intuïtieve GUI is gelijkaardig aan **Microsoft Excel**.



GNOME Office

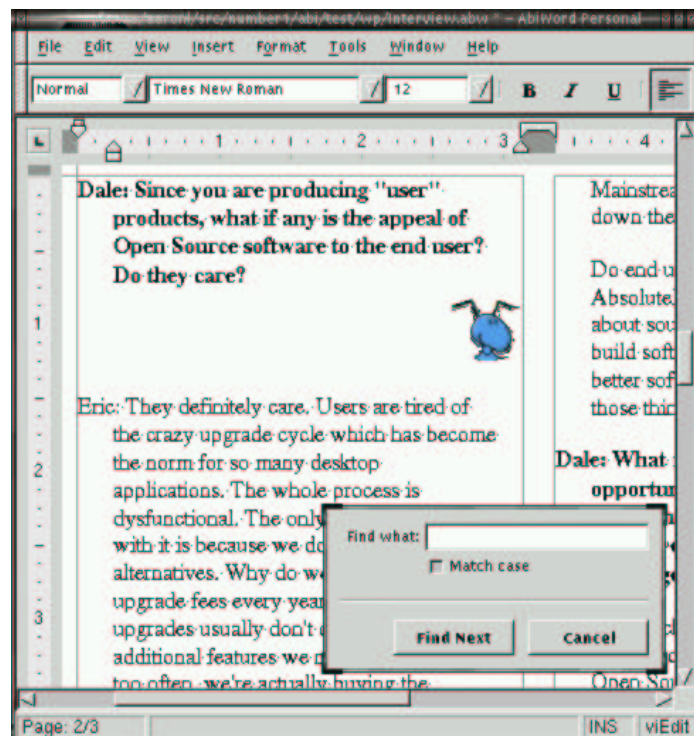
GNOME, de tegenhanger van KDE, heeft pogingen ondernomen om (net zoals KDE) een Office-suit te maken. Tot op heden zijn de pogingen om een volledige Office-suit te programmeren eigenlijk een beetje mislukt. Het enige wat er is over gebleven is **Gnumeric**, een spreadsheet. Als tekstverwerker heeft het GNOME-team **AbiWord** geadopteerd. **AbiWord** is geprogrammeerd met dezelfde toolkit als GNOME en kan bijgevolg gemakkelijk geïntegreerd worden.

Een screenshot van **Gnumeric**.

	A	B	C	D	E	F	G
1		Januari	Februari	Maart	Totaal Persoon		
2	Persoon1	400	500	600	1500		
3	Persoon2	500	200	900	1600		
4	Totaal Maand	900	700	1500	3100		

De screenshot van **AbiWord** staat hiernaast.

GNOME-Office bevat nog talloze onderdelen. Meer informatie op <http://www.gnome.org/gnome-office>



StarOffice / OpenOffice

StarOffice is dé referentie voor Office-suits op het Linuxplatform. De ondersteuning die **StarOffice** geeft is volledig vergelijkbaar met **Microsoft Office**.

Zo heeft StarOffice niet alleen alle features van zijn tegenhanger, maar kan deze ook *Microsoft* documenten lezen/bewaren.

StarOffice is niet enkel beschikbaar voor Linux, je kan het ook gebruiken op andere Unices én op *Windows*. Omdat StarOffice *gratis* wordt verdeeld is dit een ware concurrent.

Staroffice/Openoffice?

StarOffice werd een tijdje geleden gemaakt door **StarDivision** uit Duitsland. Een paar jaar terug heeft **Sun Microsystems** deze software opgekocht met de bedoeling deze verder te ontwikkelen. Omdat **Sun** niet de capaciteit had om het programma te onderhouden werd **OpenOffice** gecreëerd. **OpenOffice** is eigenlijk de originele **StarOffice** min een aantal functies (professionele spellingchecker, ...) dat uitgegeven werd met de broncode. De bedoeling van **Sun** is om regelmatig de allernieuwste versie van **OpenOffice** om te zetten naar een verkoopbaar product dat de naam **StarOffice** behoudt. **OpenOffice** is gratis en de broncode is beschikbaar. **StarOffice** is in vergelijking met **Microsoft Office** vrij goedkoop. De kostprijs is ongeveer 10 maal lager.

StarOffice heeft de naam om traag te laden (eens het geladen is werkt het vrij vlot). Dit probleem wordt nu opgelost door het OpenOffice-team door alle programma's te splitsen. StarOffice/OpenOffice vergt ook vrij veel geheugen omdat het werkt met een eigen GUI toolkit.

OpenOffice is beschikbaar via:

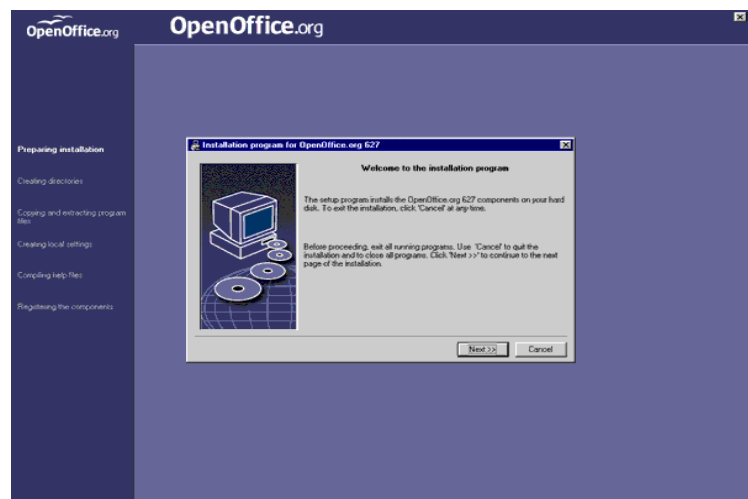
<http://www.openoffice.org>

Meer info over StarOffice kan je vinden op:

<http://www.sun.com>

Installatie

Op een **Windows** systeem zal StarOffice/OpenOffice zich installeren als een gewoon Windowsprogramma.



Op alle *Unices* heb je twee mogelijkheden:

- single user install
- network install

Een *single user install* kan je als gewone gebruiker uitvoeren. Je installeert StarOffice/Openoffice in je eigen home-folder. Het nadeel hiervan is dat je de enige gebruiker bent op het systeem die het programma kan starten. Een *single user install* is de **default** installatiemodus.

Een *network install* moet je starten als root. Deze modus maakt het mogelijk om StarOffice te installeren op een gemeenschappelijke plaats. **Vergeet niet de installatie-map te wijzigen!** Deze zal standaard op */root* staan, maar niemand kan hieruit lezen. Kies hier voor */usr/local/staroffice* of */opt/staroffice*.

Iedere gebruiker die StarOffice wil gebruiken moet dan het volgende programma starten:

```
<Dir StarOffice>/office52/program/soffice
```

Voor Openoffice (of StarOffice 6) is dit:

```
<Dir OpenOffice>/setup
```

De gebruiker krijgt dan een mini installatie zodat hij/zij persoonlijke gegevens kan aanpassen/instellen. De *network install* werkt ook als je StarOffice installeert in een aparte *share*.

Een *network install* start je met de optie *"/net"* (of in OpenOffice en StarOffice 6 met de optie *"-net"*).

De onderdelen

Star/OpenOffice is ongeveer gelijkaardig aan **Microsoft Office** indien we kijken naar de bijgeleverde programma's. In StarOffice 5.2 en lager zit er een *Desktop* bijgevoegd. Hieruit kan je nieuwe documenten opstellen. De nieuwere versies bevatten geen *Desktop* meer omdat het OS er al één heeft (het is nogal verwarrend indien je meerdere desktops hebt).

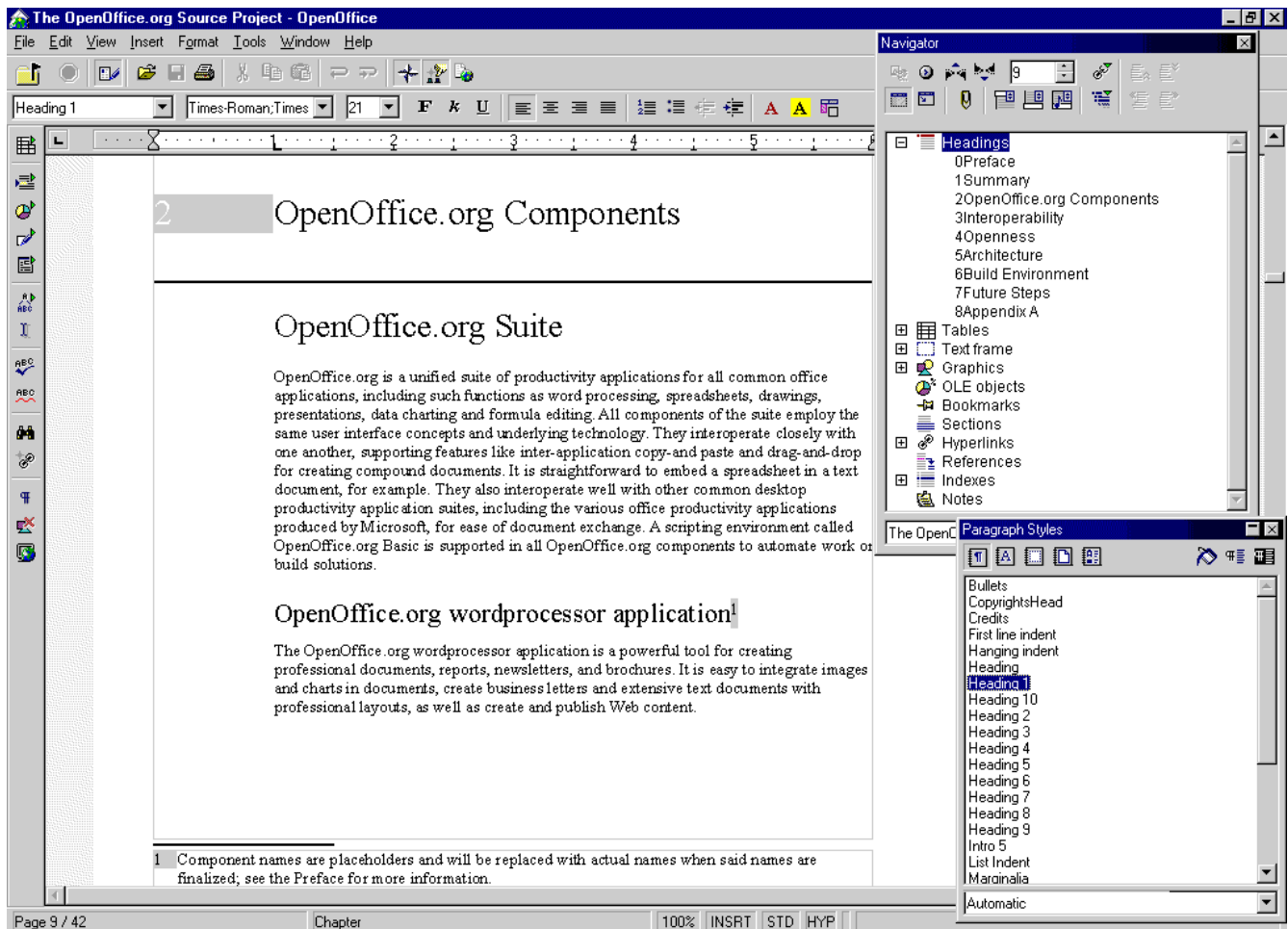
Verder zijn in OpenOffice en vanaf StarOffice 6 de e-mail en samenwerkings-applicaties verwijderd.

We gaan ieder onderdeel van nader bekijken. Deze beschrijving vormt **geen** handleiding.



Writer

Writer is het programma dat gebruikt werd om deze cursus te schrijven. De *features* zijn vrij volledig. Er is zelfs een Nederlandstalige spellingcontrole. Zowat alle documenttypes worden ondersteunt, zelfs die van **Microsoft** zodat je zonder veel problemen kan overschakelen.

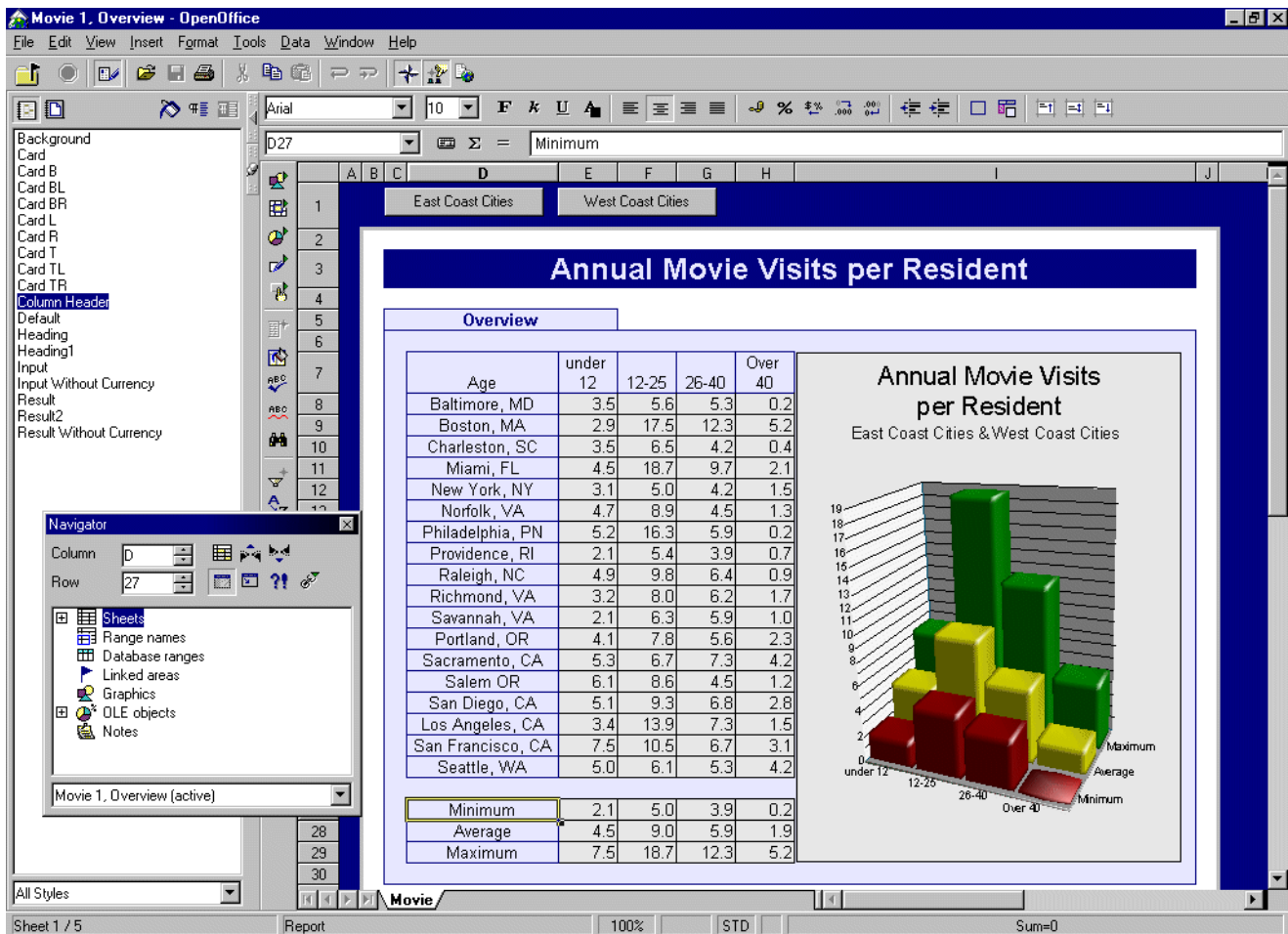


Opdracht

Open deze cursus in **Writer**

Calc

Calc ondersteunt alle bewerkingen die we ook kunnen terugvinden in **Microsoft Excel**. Net zoals **Writer** is de GUI intuïtief genoeg om onmiddellijk te beginnen. Wat vooral opvalt is dat **Calc** een zeer uitgebreide ondersteuning heeft om grafieken in 3D voor te stellen.



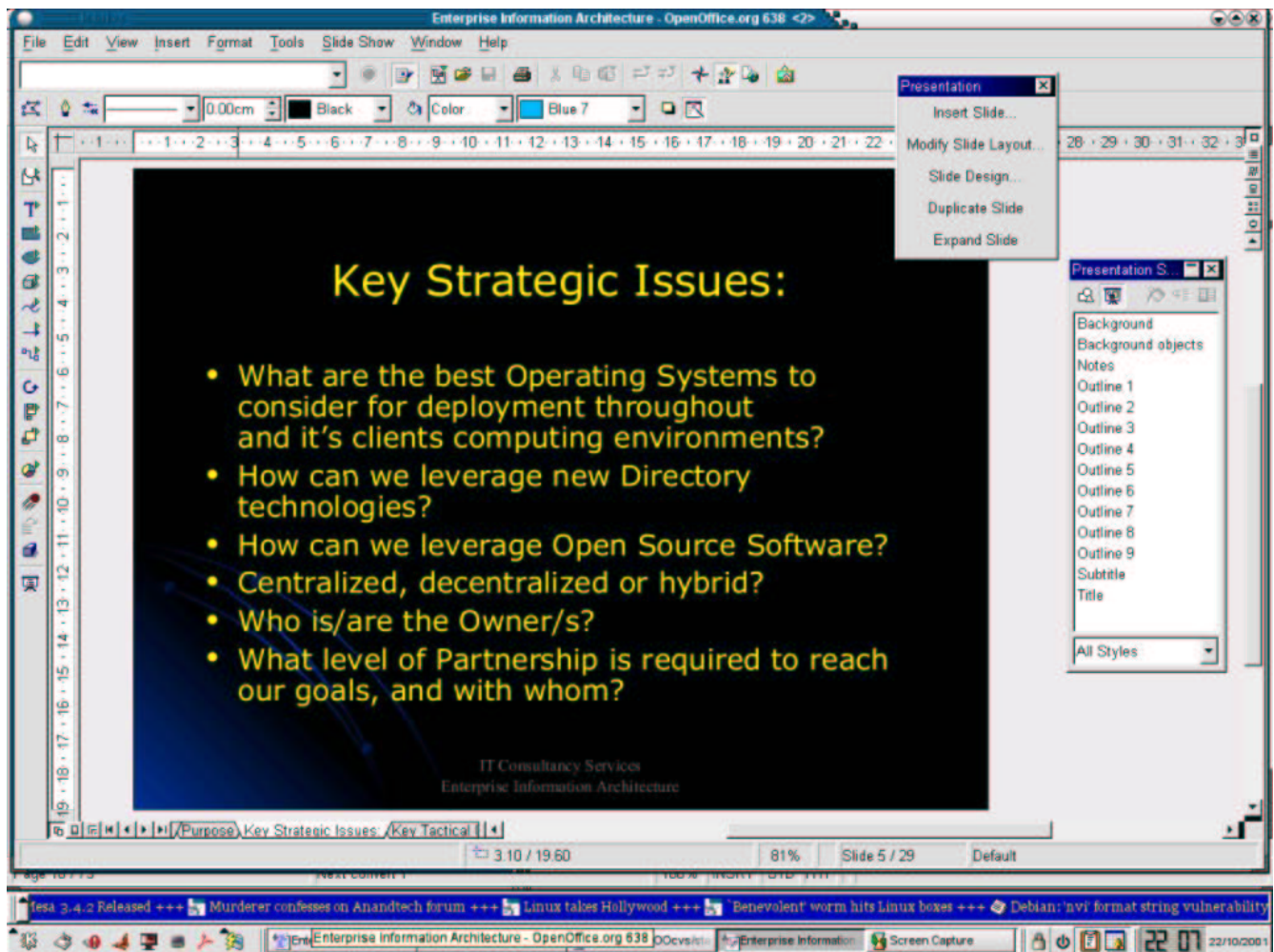
Opdracht

Probeer de spreadsheet in bovenstaande screenshot na te maken.

Cursus Linux

Impress

Impress dient om presentaties te maken. Ook hier is de interface gelijkaardig aan **Microsoft PowerPoint**.



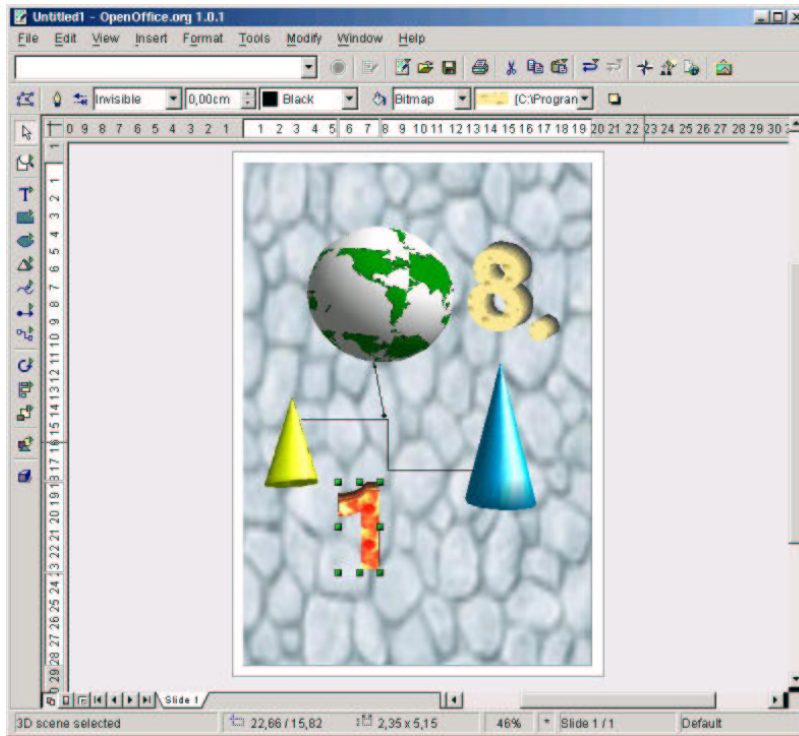
Opdracht

Maak een simpele presentatie (2-3 slides).

Draw

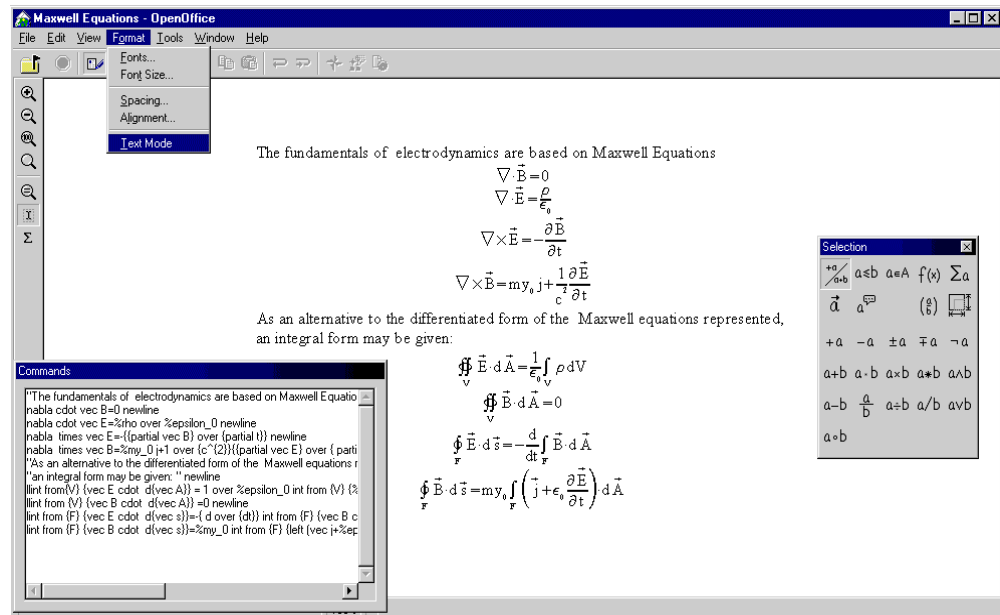
Draw is het tekenprogramma binnen Star/OpenOffice. De belangrijkste feature is de ondersteuning van 3D-componenten.

Cursus Linux



Math

Met **Math** kan je ingewikkelde wiskundige formules intypen.



Cursus Linux

StarBase (StarOffice 5.2)

StarBase is een beetje de tegenhanger van **Microsoft Access**. Zo heeft **StarBase** geen eigen *database-engine*, maar maakt gebruik van enkele externe modellen. Je kan gebruik maken van **ODBC** of **Adabas**. Als je geen database-drivers hebt geïnstalleerd kan je nog altijd de data bewaren in **dBase** formaat.

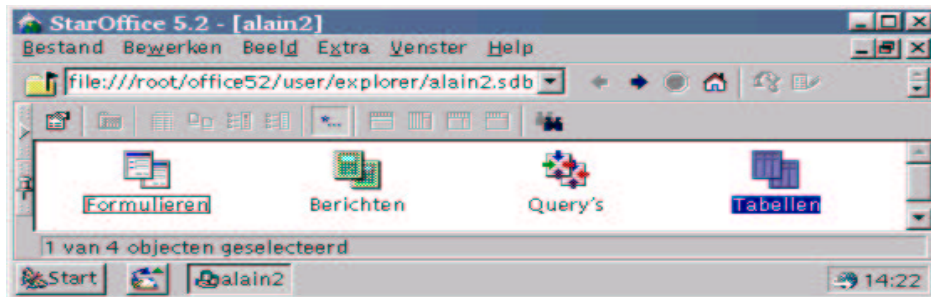
Het aanmaken van een database is minder intuïtief dan men zou verwachten.

Gebruik volgende commandoreeks om een database aan te maken:

Bestand->Nieuw->Database...

Geef deze nu een naam en specificeer (indien men niet over een externe database beschikt) voor *dBase*. Geef ook een lege directory aan waar de bestanden bewaard mogen worden (indien het een bestaande **dBase** directory is wordt deze ingelezen en de data/tabellen worden gebruikt).

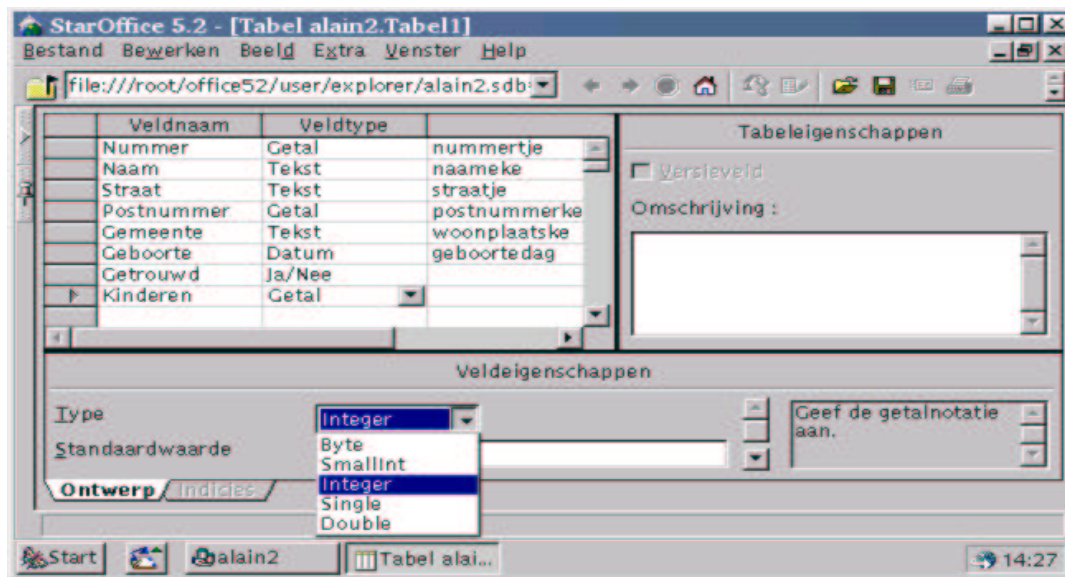
Op de **Desktop** heb je een icoon staan met de database die je hebt aangemaakt. Als je hierop klikt krijg je het volgende scherm:



Om de tabellen aan te maken gebruik je de volgende combinatie van commando's:

Rechtse klik op Tabellen->Nieuw->Tabel->Tabelontwerp

Het volgende scherm komt dan tevoorschijn:



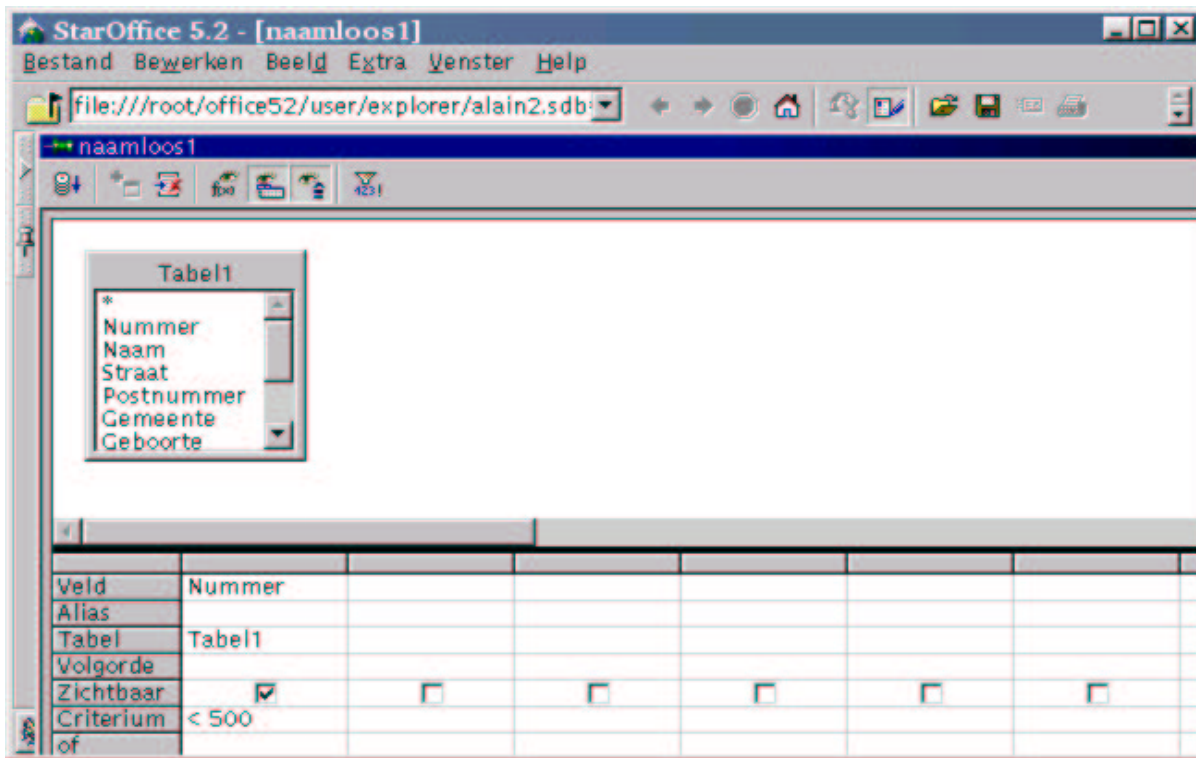
Cursus Linux

Net zoals in **Microsoft Access** kan je velden aanmaken en daarna invullen met data.

Als je *Query's* wil aanmaken ga je vanaf het database hoofdmenu (zie figuur 1) volgende commando's uitvoeren:

Rechtse klik op Query->Nieuw->Query->Queryontwerp

Je komt dan in het volgende scherm terecht:



Net zoals in **Microsoft Access** kan je Query's maken en uitvoeren.

Base (OpenOffice)

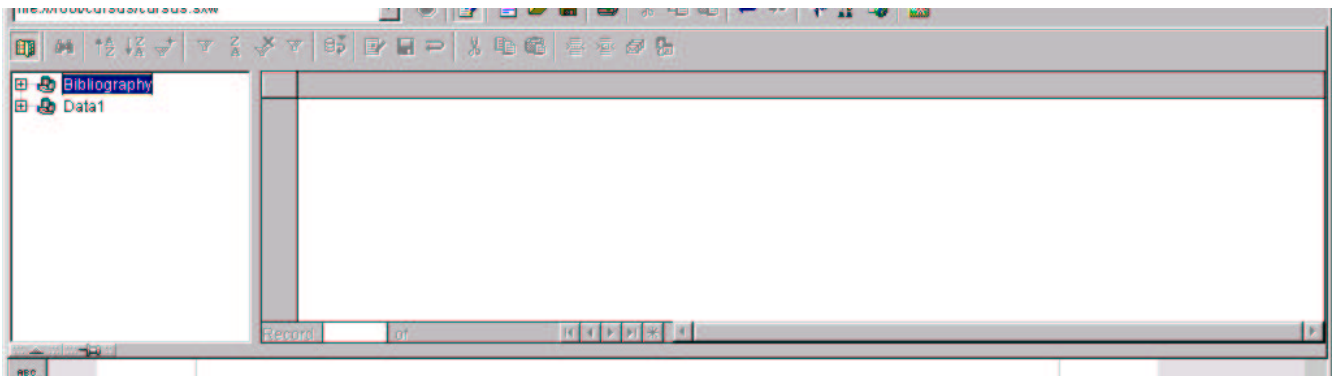
Net zoals in StarOffice is **Base**, in OpenOffice, een geïntegreerd programma. Het gebruik van deze database verschilt van de vorige versies.

 Klik op het "**Data Sources**" icoon (op de linkse werkbalk).

Bovenaan het beeld zal er een scherm verschijnen waarmee je de data kan beheren (zie screenshot).

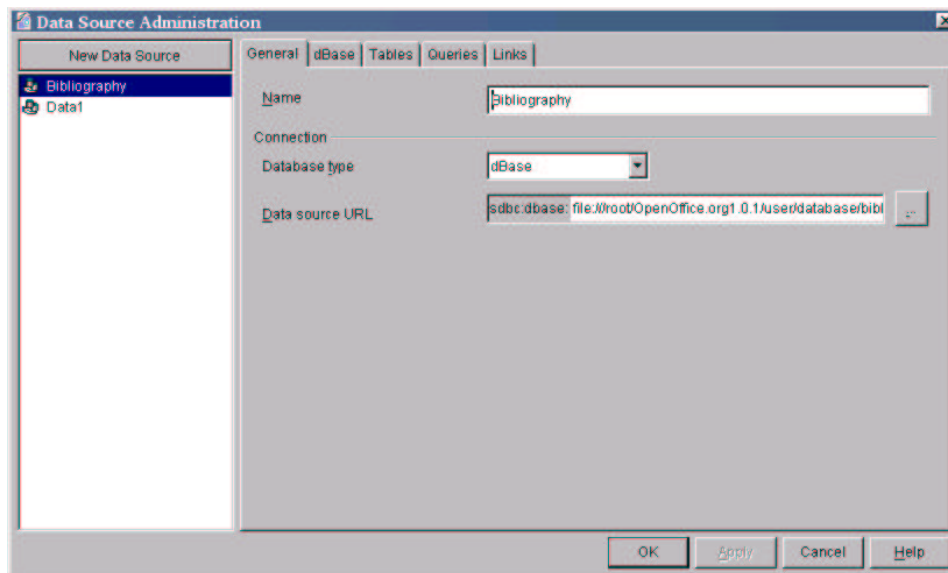
We gaan de stappen doorlopen om een database aan te maken.

Cursus Linux



Links op het scherm zie je de beschikbare database-bronnen. Door op "+" te klikken kan je op een database-bron bewerkingen maken (wijzigen/aanmaken van query's, tabellen,...). Druk in dit linkse scherm op de Rechter muistoets.

Je zal dan een menu te zien krijgen waar je "**Administrate Data Sources...**" kan kiezen (kies vervolgens deze optie).



In dit scherm kan je de beschikbare databases beheren. Maak een nieuwe database aan door op de knop "New Data Source" te klikken.

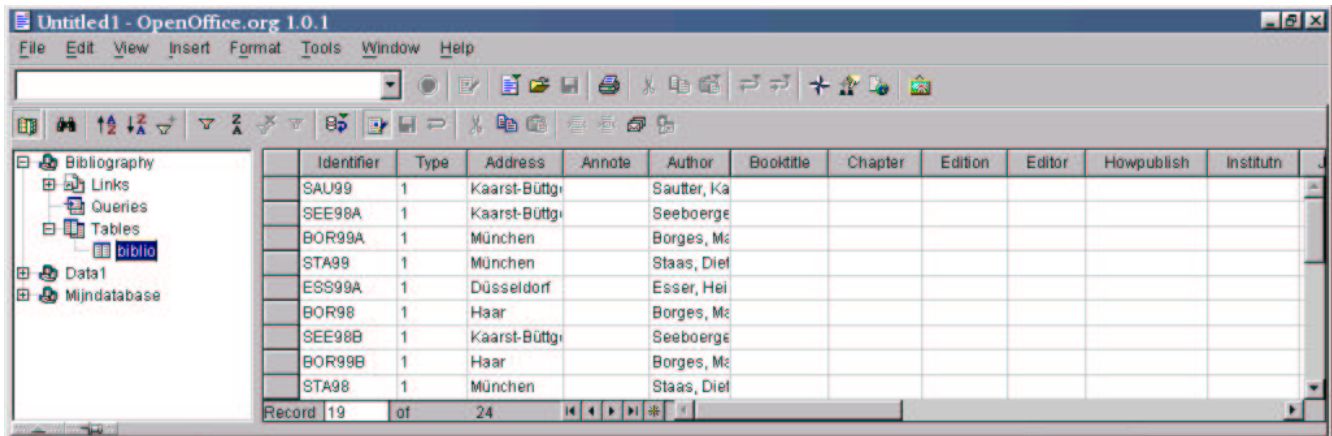
Geef de nieuwe database een naam en kies "dBase" als "**Database Type**".

Voor de "**Data Source URL**" kies je een lege map (maak eventueel gebruik van de "..." knop).

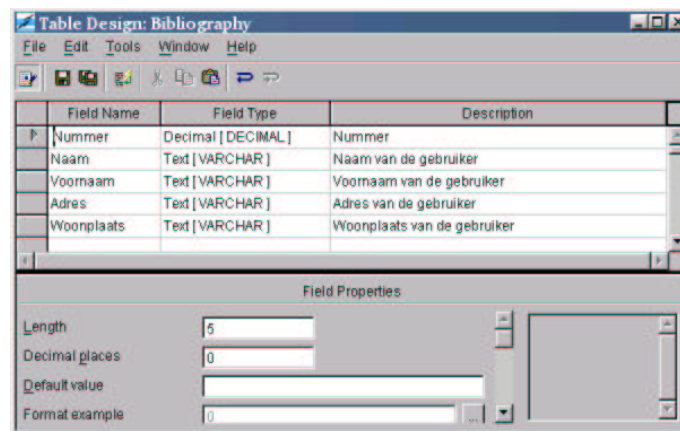
Door op "OK" te klikken kom je terug in het vorige scherm. De nieuwe database zal in de keuzelijst verschijnen.


Cursus Linux

Om een nieuwe tabel aan te maken klik je eerst op "+" van de gewenste database (gebruik de "**Bibliography**" database aangezien deze al data bevat).



Klik met de rechter muistoets op "Tables". Kies voor "New Table Design". Nu kan je een tabel aanmaken met de gewenste velden.



Druk op  om de wijzigingen te bewaren (geef de tabel dan een gepaste naam). Door in de "**Tables**" gewenste tabel te kiezen kan je in het rechter scherm data invoeren.

Met de optie "Links" kan je data-koppelingen maken met OpenOffice documenten. Met "Queries" kan je lijsten maken van gespecificeerde data. Als je hierop **rechts** klikt kan je SQL-commando's aanmaken of een lijst maken met een GUI.

Hancom Office

Dit Koreaans product is vrij populair op de Aziatische markt omdat het de mogelijkheid biedt om teksten weer te geven die van rechts naar links worden geschreven (alsook internationale karakters zoals oa. Chinese tekens).

Hancom heeft ook een draagbare versie van hun Office Suite. Deze is ingebouwd in oa. de Sharp Zaurus.

Meer info op <http://www.hancom.com>

Andere pakketten

De belangrijkste en meest populaire Office pakketten zijn in dit hoofdstuk beschreven. Daarnaast bestaan er, net zoals in Windows, nog andere programma's die deze taken kunnen vervullen.

Neem gerust eens een kijkje op <http://www.freshmeat.net> of <http://www.linuxberg.com>.

The image shows two screenshots of the Hancom Office Suite. The top screenshot is a document window titled 'mystery_snapshot.hwp' in HancomWord 6.0. It displays a document titled 'The Seven Wonders of the Ancient World' with seven images arranged in a grid. Below the images is an 'Introduction' section. The bottom screenshot is a spreadsheet window titled 'HancomSheet - Book1.xls' in HancomSheet. It shows a table with data for the Seven Wonders and a bar chart titled 'Compare with Height'.

A wonder of the world	Area(%)	Height(%)	The Time(years)
Great Pyramid	765	486	B.C.4500
Hanging gardens	480000	315	B.C.500
Statue of Zeus	17.82	36	B.C.457
Temple of Artemis	111800	60	B.C.365
Mousoleum	3097.2	150	B.C.350
Colossus		108	B.C.280
Lighthouse	5400	384	B.C.290

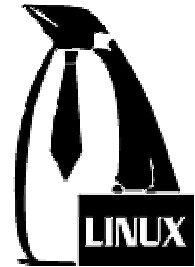
Compare with Height

Wonder	Height (%)
Lighthouse	384
Colossus	108
Mousoleum	150
Temple of Artemis	60
Statue of Zeus	36
Hanging gardens	315
Great Pyramid	486

Servers

De meest geschikte opdracht voor een linuxsysteem is deze te gebruiken als server. De performantie van een linuxserver ligt vele malen hoger dan die van andere *non-UNIX* systemen. Dit komt omdat het *bestandssysteem* en de *scheduler* geoptimaliseerd zijn om een hoge *Input/Output* aan te kunnen (zelfs met beperkte hardware).

Het is aan te raden om op servers geen X-Windows te draaien (dus runlevel 3), tenzij het absoluut noodzakelijk is.



Services

In het bestand `/etc/services` vind je een lijst van poorten, hun benaming (en eventueel commentaar). Dit bestand wordt vaak gebruikt door servers om een poortnaam te *mappen* naar een poortnummer.

(x)inetd, the superserver

Inetd is een server die tegelijkertijd naar meerdere poorten luistert. Indien er een *request* binnenkomt op een bepaalde poort wordt het correcte programma gestart om de *request* te verwerken. Door **inetd** is het mogelijk om slechts één server te draaien in plaats van meerdere, waardoor op geheugen kan bespaard worden. Veel gebruikte services draai je best niet onder **inetd** omdat deze dan telkens de externe service moet opstarten (met andere woorden: *het zal trager werken*).

Xinetd is een uitbreiding op **inetd**. Het belangrijkste verschil is dat bij **xinetd** de configuratie in een aparte map staat en dat je per *service* meer opties hebt. **Xinetd** wordt vooral gebruikt op nieuwere versies van Linux.

De *(x)inetd superserver* wordt vooral gebruikt om kleine servers zoals *telnet*, *rsh*, *time*,... te starten.

We bekijken even een gedeelte uit `/etc/inetd.conf`

```
-----  
#time    stream  tcp     nowait  root    internal  
ftp      stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd  -l  -a  
telnet   stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd  
#pop-2   stream  tcp     nowait  root    /usr/sbin/tcpd  ipop2d  
#pop-3   stream  tcp     nowait  root    /usr/sbin/tcpd  ipop3d  
#imap    stream  tcp     nowait  root    /usr/sbin/tcpd  imapd  
-----  
1        2        3        4        5        6
```

De lijnen die beginnen met "#" zijn **commentaar** of **inactieve services**.

1 = De poort waarop de *service* draait (zie */etc/services*)

2 = Het type van de informatiestroom

3 = Het protocoltype

4 = Timing

5 = Gebruiker die de *service* start

6 = Programma dat moet gestart worden bij een inkomend *request*

De configuratie van **xinetd** ziet er helemaal anders uit. Als je een *listing* opvraagt van de directory */etc/xinetd.d* zie je een heleboel bestanden. Elk bestand is een *service*.

We bekijken een voorbeeld van zo'n bestand:

```
service telnet
{
    flags                = REUSE
    socket_type          = stream
    wait                 = no
    user                 = root
    server               = /usr/sbin/in.telnetd
    log_on_failure       += USERID
    disable              = no
}
```

We vinden ongeveer dezelfde data terug als bij **inetd**. We kunnen hier nog parameters aan toevoegen die door **inetd** niet ondersteund worden.

Als je iets gewijzigd hebt kan je de aanpassingen activeren door **(x)inetd** te herstarten met het commando:

Inetd: **/etc/init.d/inet restart**

Xinetd: **/etc/init.d/xinetd restart**

Nota: Het is aan te raden om enkel *services* te laten draaien waarvan je zeker bent dat je ze nodig hebt voor het functioneren van het systeem.

Mailserver

De meest gebruikte mailservers op het internet is **sendmail**. Sendmail zit standaard bij iedere Linuxdistributie. Sommige distributies leveren ook mailservers zoals **postfix** en **qmail**. We gaan enkel de configuratie van sendmail bespreken.

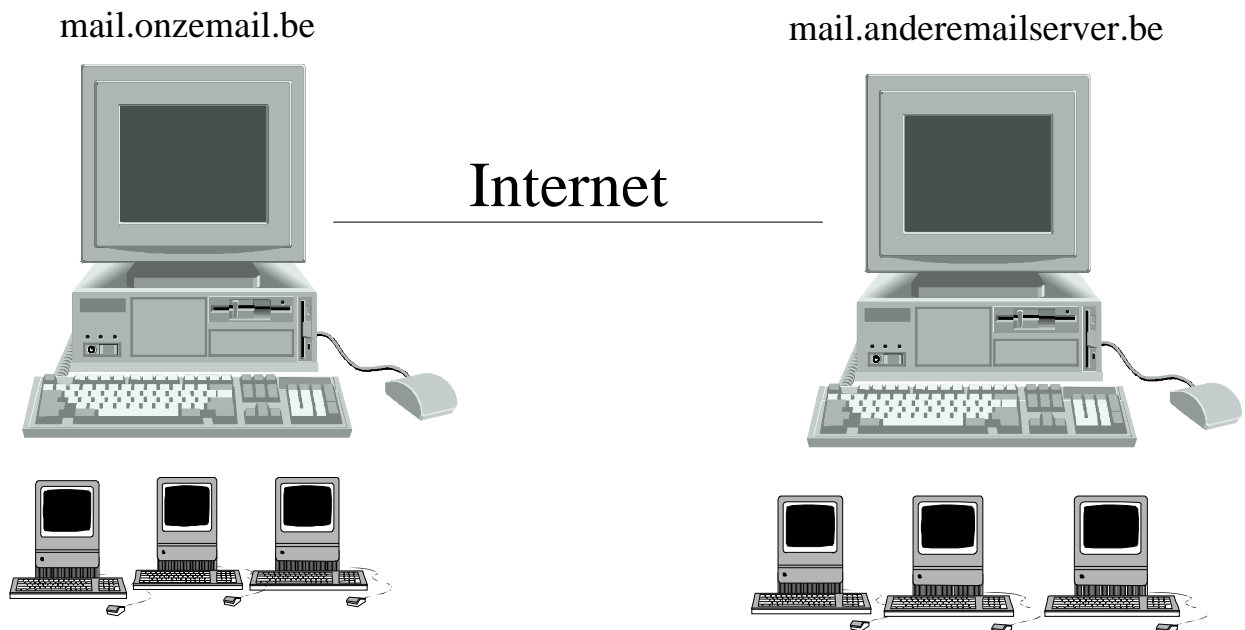


De configuratie van sendmail is enorm ingewikkeld indien je deze manueel wil doen. We verkiezen bijgevolg om onze server te configureren met **linuxconf**.

Verzenden van mail

Iedere machine heeft een domeinnaam (zelfs indien de PC niet is verbonden met het internet). De machine zal dan enkel mail aanvaarden voor jouw domein. Indien er naar een ander domain e-mail wordt verstuurd dan zal onze mailservers gaan zoeken welke computer e-mail ontvangt voor dat bepaald domein (dit is de beruchte MX-record in een DNS configuratie). Als de server bestaat wordt het bericht doorgezonden naar die server (die in principe ook enkel mail wil ontvangen voor zijn eigen domain). Andersom worden alle mails naar ons domein op onze mailservers geplaatst door andere mailservers die een bericht willen sturen.

De berichten worden gepost op SMTP poort 25 van de "remote host".



Cursus Linux

Indien we achter een andere mailserver zitten (en er is geen rechtstreekse internetverbinding, of poort SMTP 25 is geblokkeerd) kunnen we al de mail die we willen versturen naar een **relay** zenden (dit is meestal de mailserver van jouw Internet Service Provider). De **relay** zorgt er dan wel voor dat onze mail op de juiste bestemming aankomt. De **relay** wordt in de configuratie ook soms **smart host** genoemd.

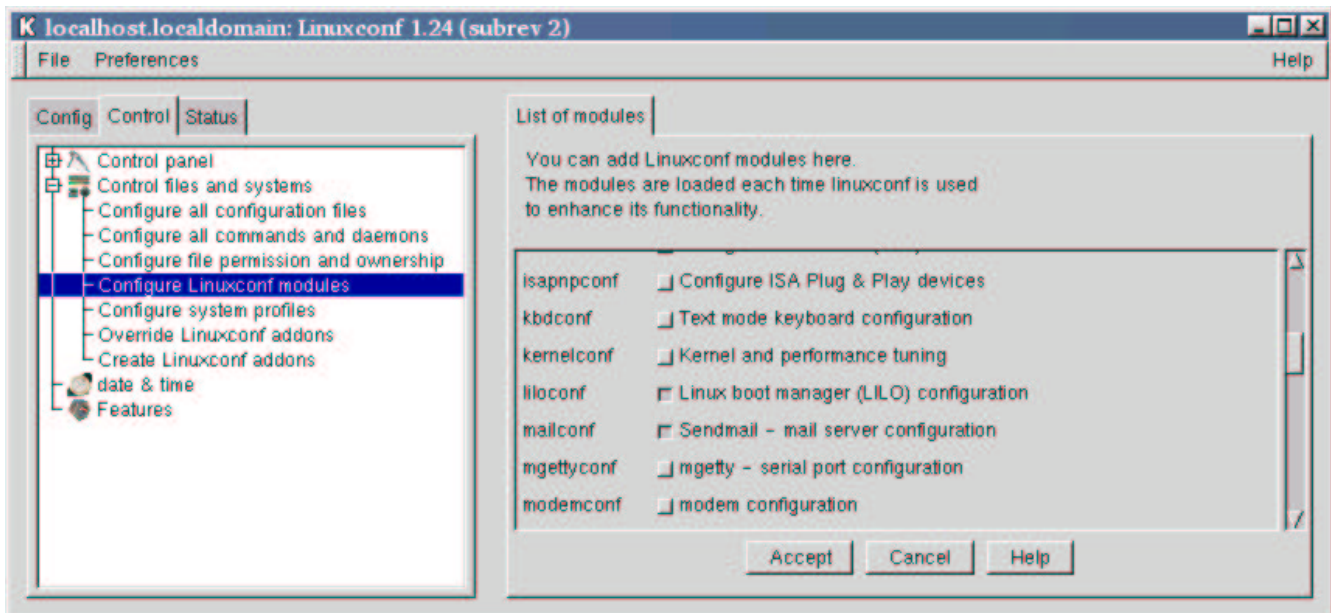
Onze eigen server moet natuurlijk ook de mail **relayen** (mails die van e-mail-clients komen). Als vuistregel kan je stellen dat je de relaying mag openzetten voor adressen op het interne netwerk (de mensen die jouw mailserver gebruiken).

Iedere gebruiker op het systeem krijgt een e-mail adres. We kunnen ook *accounts* aanmaken die enkel kunnen *mailen* (*POP Only accounts* in **linuxconf**).

Je kan ook "*aliases*" maken. Dit zijn virtuele gebruikers die niet gekend zijn in het systeem, maar wel door de mailserver. Zo'n virtuele gebruiker komt overeen met één of meerdere mailgebruikers.

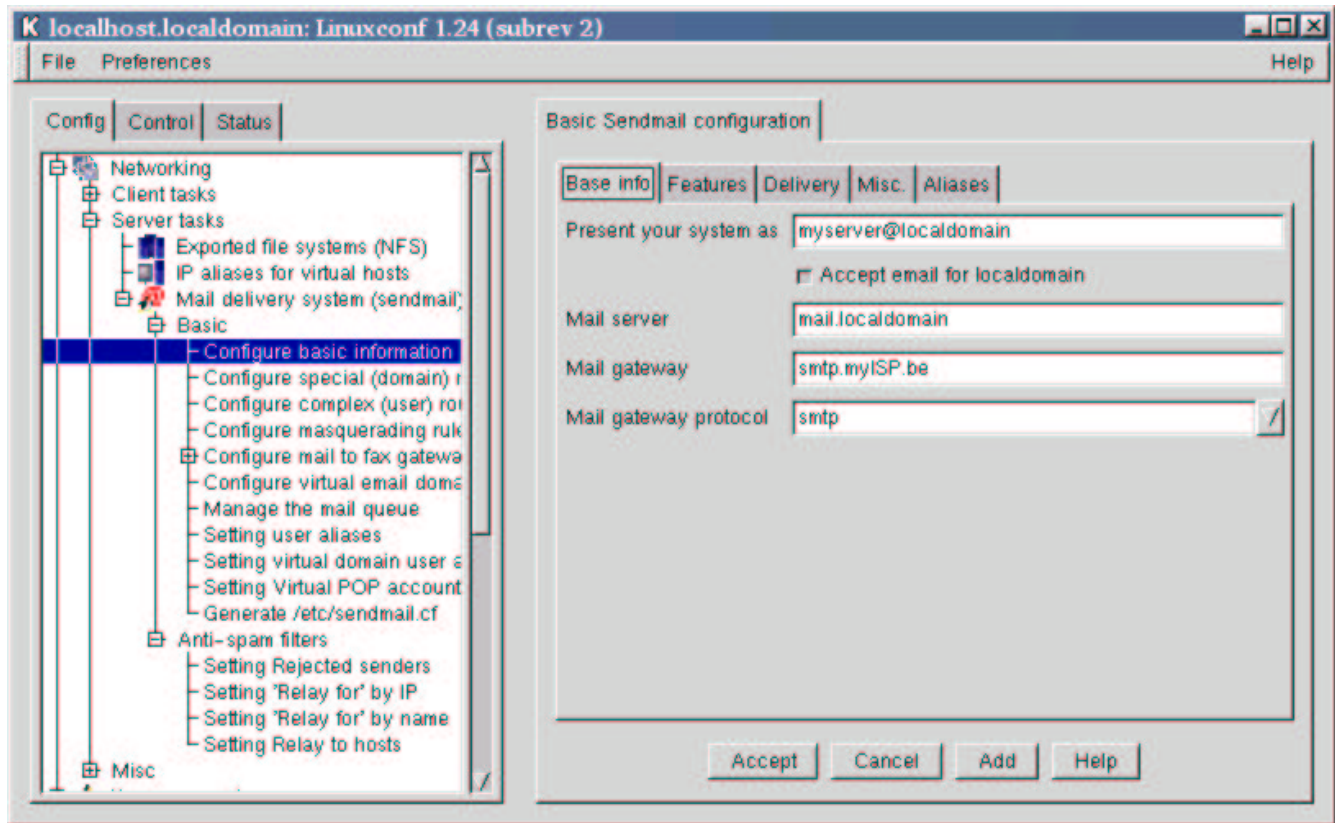
We kijken even naar een voorbeeld configuratie van **sendmail** in **linuxconf**.

Allereerst configureren we **linuxconf** zodat we ook de mailconfiguratie kunnen zien en gebruiken:



Na de wijziging moet je **linuxconf** afsluiten en herstarten.

Je zal de configuratie van sendmail vinden in **tab Config**, **tree Networking, Server tasks, Mail delivery system (sendmail)**.

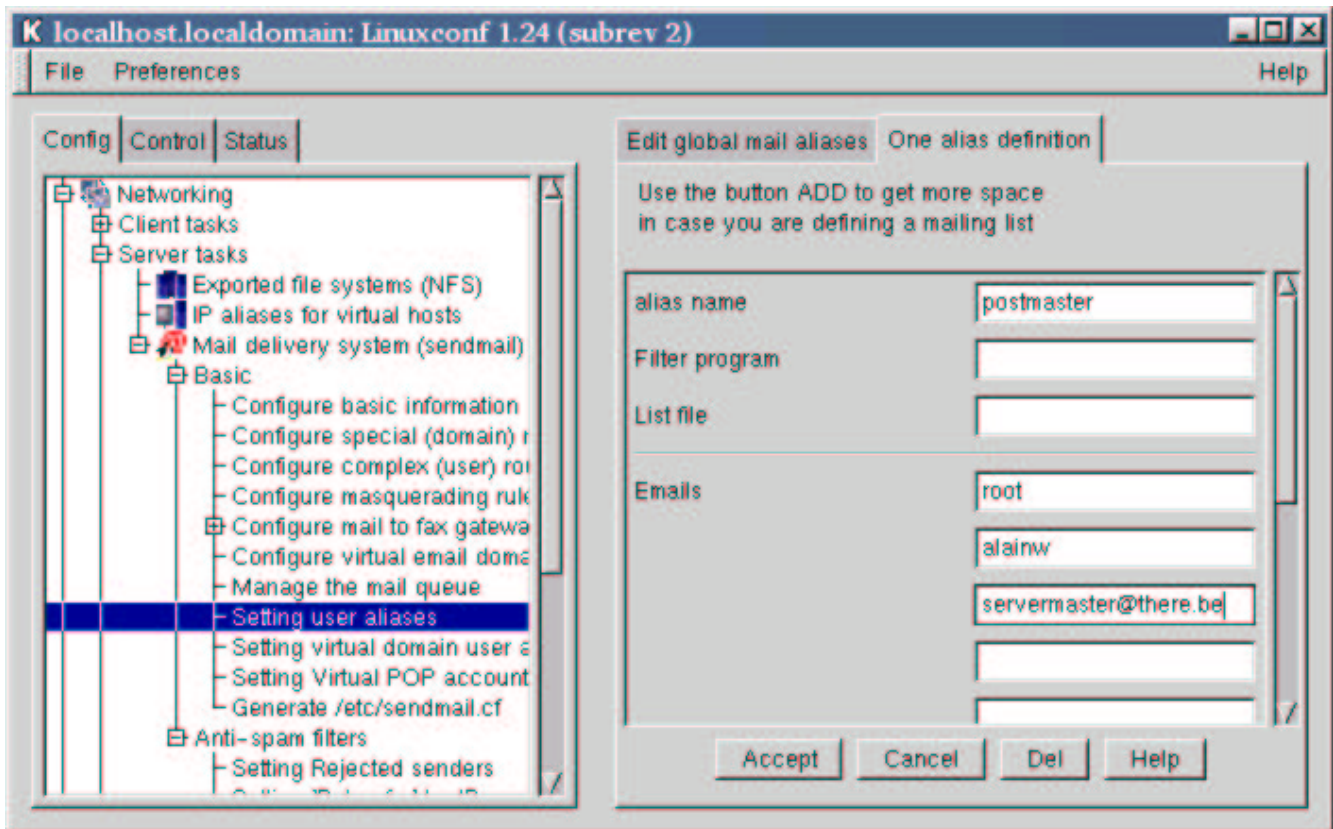


We gaan kort eens de belangrijkste instellingen overlopen:

- **Present system as:** volledige domeinnaam van het systeem
- **Accept E-mail from ...:** aanvaardt inkomenden e-mail van andere systemen
- **Mail server:** geeft de mogelijkheid om de mail op te slaan op een externe machine
- **Mail Gateway:** stuurt al de externe mail door naar deze host
- **Enable Relay Control:** controle of we de binnenkomende mail willen aanvaarden of doorsturen. Het instellen gebeurt in de *Anti-spam filters*-boom.

Meer uitleg van alle velden kan je oproepen via de *Help* knop.

Gebruikersaliases kan men aanmaken om distributielijsten te genereren. Een distributielijst is één e-mailadres voor een groep gebruikers (binnen het systeem wordt de binnengekomen e-mail *geforward* naar de correcte ontvangstadressen). Je kan een *alias* ook gebruiken om een bepaalde gebruiker meerdere e-mailadressen binnen het systeem te geven en dat toch alle mail in één enkele mailbox terecht komt.



De *Anti-Spam filters* zorgen ervoor dat je kan beheren welke computers mail mogen *posten* op jouw server.

Na het aanpassen van de configuratie zal **linuxconf** het configuratiebestand genereren. Herstart de **sendmail** daemon met het commando `/etc/init.d/sendmail restart`

Nota: Het is mogelijk om de configuratie actief te maken zonder de mailserver te herstarten. Een "`killall -1 sendmail`" is hier voldoende. Voor *non-critical* servers is een herstart wel aan te raden.

Command line

Je kan de instellingen ook wijzigen door de configuratiebestanden aan te passen. De leesbare configuratie vind je in `/etc/mail`.

De bestanden los van de mailserverconfiguratie en die dynamisch kunnen zijn (zoals bijvoorbeeld de *anti-spam filter*) wijzig je in de respectievelijke bestanden. We kijken eens naar `/etc/mail/access` (hosts waarvoor we gaan *relayen*):

```
localhost          RELAY
127.0.0.1          RELAY
```


Deze informatie wordt door **sendmail** uit *databasebestanden* gelezen (wegens performantie). Daarvoor moeten we deze bestanden *compileren*. We vinden in deze directory een **Makefile** die we kunnen gebruiken en die ervoor zorgt dat alle configuratiebestanden herbouwd worden. Op die manier gebruik je dus het "**make**" commando die de **Makefile** uitvoert. De onderliggende commando's zien er zo uit:

```
makemap -f hash access.db < access
```

De instellingen van de mailserver zelf worden bijgehouden in *"/etc/mail/sendmail.mc"*. Je zal merken dat er opties in vermeld staan die lijken op de benamingen in **linuxconf**, maar de syntax lijkt vrij cryptisch (je mag het bestand gerust eens bekijken). Dit bestand is nog niet cryptisch genoeg voor **sendmail**, daarvoor moeten we dit bestand compileren met het commando:

```
m4 /etc/mail/sendmail.mc > /etc/sendmail.cf
```

Probeer nu eens wijs te geraken uit *"/etc/sendmail.cf:-)*

De *aliases* kan je aanpassen in het bestand *"/etc/aliases"*.

Kort voorbeeld van de syntax:

```
root:          marc
bogusgebruiker: jef
everyone:      root marc jef qtuser
```

De eerste regel zegt dat alle mail naar *root* doorgestuurd moet worden naar *marc*. Indien er e-mail binnenkomt naar het adres *bogusgebruiker*, dan wordt dit *forward* naar *jef*. Als er een e-mail ontvangen wordt voor *everyone*, dan ontvangen zowel *root*, *marc*, *jef* en *qtuser* dit bericht.

Na het aanpassen van de *alias* instellingen moet je het programma "**newaliases**" starten om de veranderingen actief te maken.

Nota: De *destination*adressen mogen *échte* mailadressen zijn (vb: *foo@bar.be*)

Ontvangen van mail

Binnengekomen berichten worden op de server opgeslagen onder *"/var/spool/mail"*. Iedere gebruiker heeft daar een eigen bestand. Het bestand wordt gelezen indien je een e-mail cliënt start.

In normale omstandigheden gaat men niet willen dat de gebruikers op de server moeten aanloggen om hun mail te lezen. Je kan op de *remote*-werkstations gebruik maken van een e-mail cliënt (zoals bijvoorbeeld *Outlook* of *Netscape Messenger*). Op de server plaats je een bijkomende *daemon* zoals **imap** of **pop**. Bij RedHat vindt je deze *daemons* in de **imap RPM**. Na de installatie van deze RPM kan je in **(x)inetd** de servers die je wil gebruiken activeren. De gebruikers kunnen dan hun e-mail afhalen op de server (mits gebruik van hun *loginnaam* + *paswoord*).

De keuze of je **imap** of **pop** gebruikt hangt van volgende elementen af:

pop

Handig als je de e-mails wil laten downloaden door de gebruikers. Afgehaalde mail wordt op de server verwijderd. De gebruiker is zelf verantwoordelijk voor het bijhouden van zijn binnengekomen berichten.

Dit is het systeem dat de meeste *Internet Service Providers* toepassen

imap

De binnengekomen mails blijven op de server. De gebruiker kan de e-mails zien en opvragen.

Nota: Dit is enkel voor het opvragen van e-mail. Het verzenden van e-mail door de cliënt gebeurt nog altijd via SMTP op poort 25 (dus zorg ervoor dat je *relaying* voor deze hosts toelaat).

Andere mailservers

Er bestaan naast **sendmail** nog enkele andere mailservers:

- **Qmail:** gemaakt om **sendmail** te vervangen, de maker beweert dat **qmail** een veel betere beveiliging heeft. **Qmail** is *open source*.
- **Postfix:** deels gesponsord door *IBM* is het net zoals **Qmail** een alternatief voor **sendmail**. Dit programma is ook *open source*.
- **SamsungContact:** commerciële mailserver van *Hewlett Packard*. Deze wil een volledig alternatief zijn voor *Microsoft Exchange*. (<http://www.samsungcontact.com>).
- **Bynari:** commerciële mailserver. Volledige integratie met *Microsoft Exchange* is mogelijk (<http://www.bynari.net>).

Er zijn nog meerdere pakketten voorhanden, maar in de meeste gevallen gaat dit om een bundeling van **sendmail/qmail/postfix** met handige configuratietools (een voorbeeld is de *SuSE email server*).

Apache Webserver

Apache is de meest succesvolle webserver op het internet. Volgens <http://www.netcraft.co.uk> draaien meer dan 60% van alle webservers deze software. **Apache** is ook platformonafhankelijk, dit maakt het mogelijk om het op alle UNIX en non-UNIX systemen te gebruiken (er bestaat een Apache versie voor *Windows*). De naam van deze server is ontstaan omdat er iemand de NCSA server (een webserver uit de beginjaren van het internet) heeft verbeterd door er *patches* voor te maken. Daaruit is de naam **A Patchy Server** ontstaan. Dit is later veranderd in **Apache**.



Er zijn GUI tools voorhanden om een Apache webserver in te stellen, maar in de meeste gevallen zijn deze niet meegeleverd bij de distributie. Configuratie is trouwens niet altijd nodig.

Naast het sturen van statische (html) bestanden kan Apache ook dynamische scripting aan. Dit kan in de vorm van CGI (scripts of binaries), PHP (scripting in webpagina's), JAVA servlets,... CGI en PHP ondersteuning zitten er in bijna alle gevallen ook bij.

De plaats waar je de webbestanden onderbrengt is verschillend van distributie, bij

	<u>\leq RH 6.2</u>	<u>\geq RH 7.0</u>
CGI	/home/httpd/cgi-bin	/var/www/cgi-bin
HTML+PHP	/home/httpd/html	/var/www/html

De plaats van de bestanden kan gewijzigd worden in het configuratiebestand.

We overlopen even de belangrijkste instellingen in */etc/httpd/conf/httpd.conf* (in het configuratiebestand staat commentaar geschreven die duidelijk genoeg is).

```
MinSpareServers 1
MaxSpareServers 5
StartServers 4
MaxClients 150
```

Dit slaat op performantietuning. We definiëren hoeveel connecties er mogen aanvaard worden en hoeveel connecties we verwachten te verwerken.

```
Listen 8080
```

Luister naar poort 8080. Standaard luistert **Apache** op poort 80.

```
User apache
Group apache
```

Start de server onder deze gebruiker. We willen niet dat **Apache** tot alles toegang heeft.

```
ServerName localhost
```

Hierin zet je best de naam van jouw PC (*localhost* als naam mag, maar is niet aan te raden).

```
DocumentRoot "/home/httpd/html "
```

De directory waar de website inzit.

```
Alias /doc/ /usr/share/doc/  
<Location /doc>  
    order deny,allow  
    deny from all  
    allow from localhost  
    Options Indexes FollowSymLinks  
</Location>
```

Met deze richtlijnen wordt de map `/usr/share/doc` door de webserver aangeboden als <http://www.myserver.be/doc>.

```
DirectoryIndex index.html index.htm index.php
```

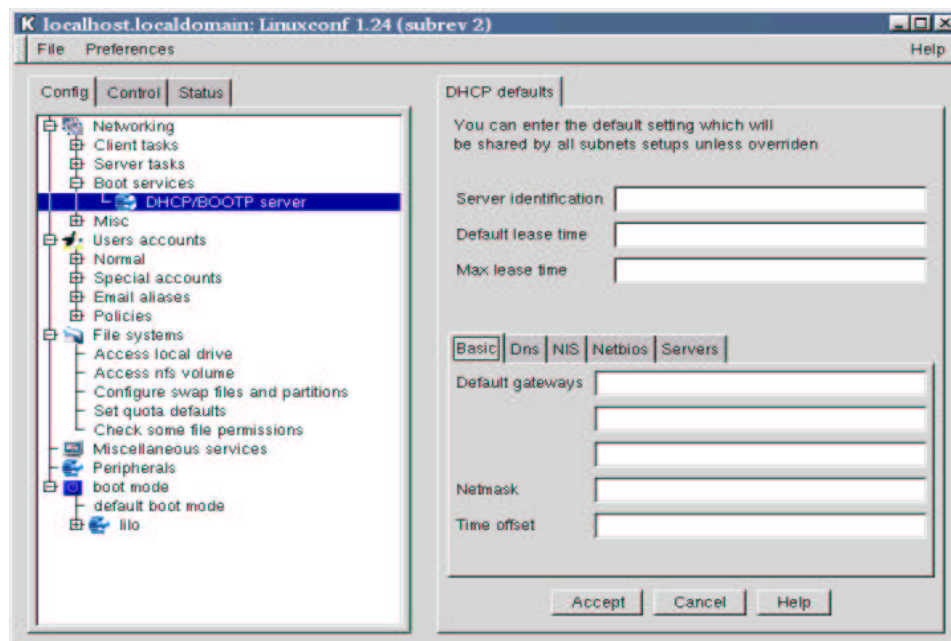
Indien er geen bestandsnaam is opgegeven zal de server in de volgorde van de aangegeven namen de bestanden openen. Vindt hij deze bestanden niet, dan krijg je een foutmelding.

Het startscript van Apache is `"/etc/init.d/httpd"`.

DHCP server

Je kan op een dynamische manier IP-adressen uitdelen aan werkstations/andere servers door het opzetten van een DHCP server. Je moet dan geen boekhouding meer bijhouden welke computers bij welke IP-configuraties passen.

Je kan best de DHCP daemon instellen via **linuxconf** (vergeet de module niet aan te zetten!).



Het bestand */etc/dhcpd.conf* wordt dan weggeschreven. Je mag deze met de hand aanpassen of zelf schrijven. De syntax is vrij eenvoudig te begrijpen.

Een voorbeeld van een configuratiebestand:

```
server-identifier bigblue.dyndns.org;
default-lease-time 86400;
max-lease-time 259200;
option domain-name "telenet.be";
subnet 192.168.1.0 netmask 255.255.255.0{
    range 192.168.1.100 192.168.1.250;
    default-lease-time 86400;
    max-lease-time 259200;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1;
    option domain-name "pandora.be";
}
```

De daemon starten kan je met het */etc/init.d/dhcpd* script.

Troubleshoot DHCP server (Common errors)

- Indien de DHCP-daemon niet start kan het zijn omdat je nog even een leeg *lease* bestand moet aanmaken in */var/state/dhcp*. Gebruik het commando "**touch /var/state/dhcp/dhcpd.leases**".
- Als de DHCP server adressen moet uitdelen op een netwerk dat aan een andere netwerkkaart hangt dan *eth0*, dan moet je even het startscript */etc/init.d/dhcpd* aanpassen. Verander dan (ingeval DHCP op *eth1* moet draaien) de lijn "*daemon /usr/sbin/dhcpd*" in "*daemon /usr/sbin/dhcpd eth1*".
- DHCP broadcasts gaan niet over *routers*.
- Indien je het probleem hebt dat je specifiek geen adressen kunt leveren aan *Windows9x* computers, probeer dan eens het commando "*route add -host 255.255.255.255 dev eth0*" (verander *eth0* in de naam van de interface waarop je DHCP levert). Indien dit werkt kan je het commando toevoegen aan "*/etc/rc.d/rc.local*".

Nieuwere versies lossen dit probleem intern op (of maken automatisch deze route aan)

DNS

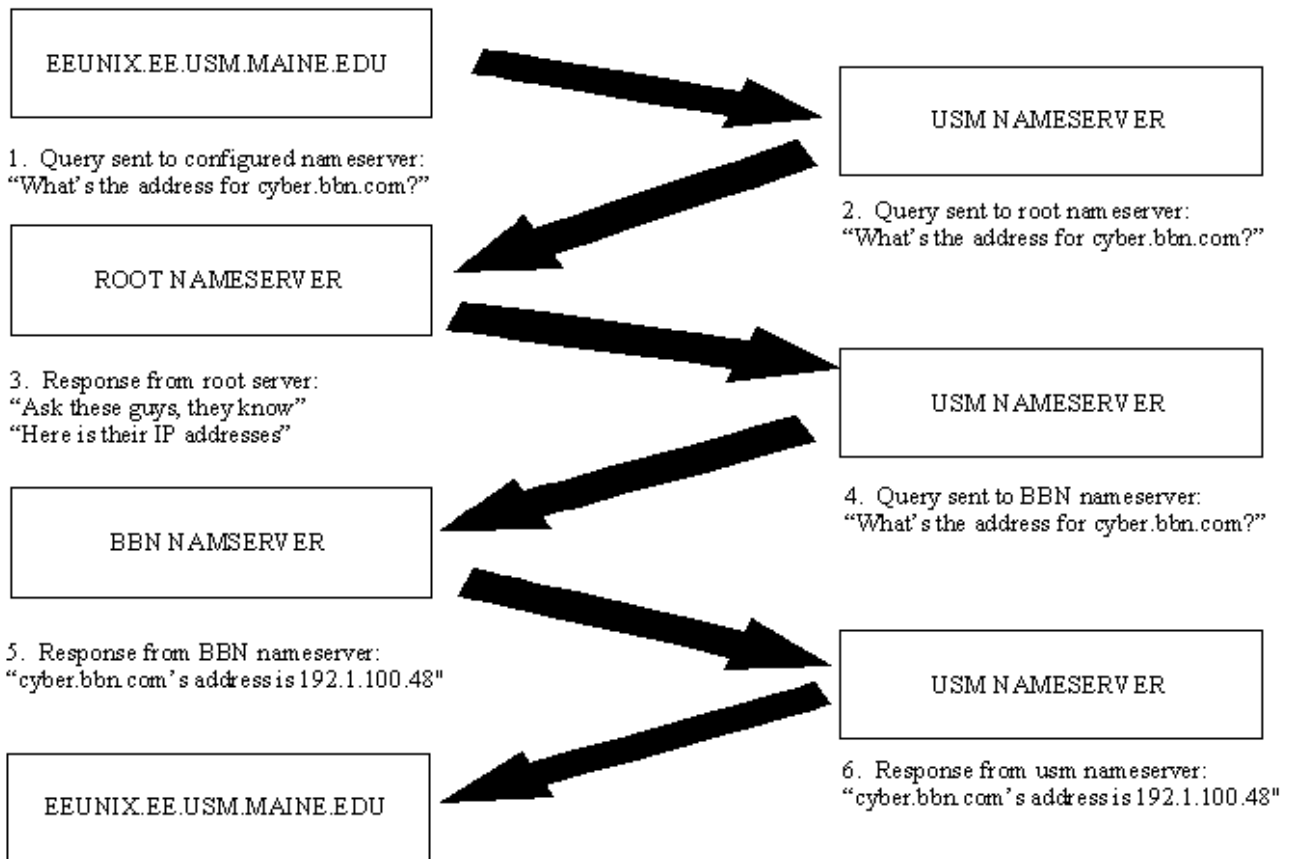
DNS staat voor *Domain Name Service*. Met deze service kunnen we computernamen omzetten in IP-adressen (en omgekeerd). Het opzetten van een DNS server is een nachtmerrie (op eender welk OS of systeem). Dus het is aangeraden om een grafische setup-tool zoals **linuxconf** hiervoor te gebruiken of de DNS howto erbij te nemen.

<http://www.linuxdoc.org> (DNS-howto)

<http://www.isc.org/products/BIND/>

<http://www.ripe.net/ripe/docs/ripe-192.html>

Werking



Query's worden uitgevoerd op poort 53 (zowel TCP als UDP)

Configuratiemogelijkheden

- **Caching-only:** steunt op de antwoorden van andere servers
- **Primary:** Er staat een bestand op de server met de DNS gegevens van het domein waar deze server "authoritative" voor is.
- **Secondary:** De server heeft een kopie-bestand van de *primary*.

Caching server

Het instellen van een "caching-nameserver", welke de meest gebruikte configuratie is, kan vrij gemakkelijk. Het is voldoende om (naast de DNS server zelf) de **caching-nameserver RPM** te installeren vanaf de CD. Vanaf dat moment kan de server de DNS-query's opvragen bij een aantal ingestelde DNS *root-servers* (root-server: DNS servers boven het .be/com/... domain).

Je moet nog wel instellen welke computers DNS-query's mogen aanvragen. Dit doe je in */etc/named.conf* in de *options* sectie.

```
allow-query {
    192.168.1.0/24;
};
```

Vaak is het wenselijk dat we de DNS-servers van onze *Internet Service Provider* gebruiken omdat dit sneller is. We kunnen dit oplossen door een aantal extra lijnen in */etc/named.conf* in de *options* sectie (voorbeeld met de Telenet DNS servers):

```
forward only;
forwarders {
    195.130.132.17;
    195.130.132.18;
};
```

Het script om de DNS server te beheren vind je in "*/etc/init.d/named*".

Nota: Om een Linuxsysteem in te stellen om een bepaalde DNS server te gebruiken moet je */etc/resolv.conf* aanpassen. Bijvoorbeeld:

```
nameserver 62.235.14.4
nameserver 212.35.2.2
```

Je kan dit ook instellen via **linuxconf**. De configuratie gebeurt automatisch indien men DHCP-client is.

Security

Linux is "by default" vrij goed beveiligd voor zowel *local* als *remote exploits*. Beveiligingslekken doen zich meestal voor op het programmaniveau. Een server is immers even veilig als zijn zwakste schakel.

Local exploits zijn beveiligingslekken waarbij een normale gebruiker op één of andere manier er in slaagt iets te doen op het systeem waarvoor hij/zij geen rechten heeft. Deze vorm is minder erg omdat lokale gebruikers niet anoniem zijn (men kan gemakkelijk de schuldige opsporen). De meeste systemen worden ook maar gebruikt door één persoon of meerdere personen die elkaar vertrouwen.

Deze vorm van *exploit* is enkel belangrijk indien men lokale gebruikers heeft die men niet kan vertrouwen.

Remote exploits zijn aanvallen op netwerkservices van een computer. Men probeert door bufferoverflows of programmafouten toegang te krijgen tot het systeem. Deze vorm van *exploits* zijn heel belangrijk voor computers die verbonden zijn met het internet (of een ander netwerk waar men de *remote* computers niet kan vertrouwen). Zit je in een lokaal netwerk waar je de anderen kan vertrouwen, dan moet je je hierover ook geen zorgen maken.

De grootste bron van beveiligingslekken bevindt zich in de *services*. Er is niet veel dat een *exploit* op een *service* met beveiligingslek kan tegenhouden. Er zijn wel enkele dingen die kunnen helpen:

- Gebruik enkel *services* die écht nodig zijn (zet de andere af via bijvoorbeeld **Linuxconf** → **Control** → **ControlPanel** → **Control Service Activity**). Hoe minder *services* je hebt draaien hoe gemakkelijker het is om te *patchen/upgraden*.
- De *services* die je hebt draaien in het oog houden. Indien er een beveiligingslek wordt ontdekt onmiddellijk *patchen/upgraden*.
- In de recentere Linuxdistributies draaien *remote services* meestal als een gebruiker met weinig of geen rechten. Op die manier zal als de *service* wordt gekraakt de kraker niet meer toegang krijgen dan de *service* zelf (rechten waarmee een kraker niet veel kan mee doen dus). Indien je zelf programma's schrijft/compileert, denk dan hieraan.



Indien *local exploits* belangrijk zijn voor jouw systeem, volg dan net zoals bij de *remote exploits* de **bugs&patches**.

BELANGRIJK !

Kies altijd voor moeilijk te raden paswoorden!

De meeste inbraken gebeuren nog altijd via slecht gekozen paswoorden. Gebruik dus nooit jouw geboortedatum, de naam van vriend(in), hond,... als paswoord. Maak liefst gebruik van paswoorden waarin cijfers en speciale tekens zitten.

Firewall

Een bijkomend onderdeel in de beveiliging van een computer is een *firewall*. Hiermee kan je de beveiliging van de netwerktoegang beheren. Merk wel op dat een *firewall* in geen enkel geval een goede reden is om onveilige *services* te draaien! Indien je iemand toelating geeft om een *service* met een beveiligingsfout te gebruiken, dan zal de *firewall* tot niets dienen.

Met een *firewall* kan je wel services afschermen van bepaalde computers of netwerken. Je kan ook met behulp van *routing* een beveiliging maken die een intern netwerk afschermd van de buitenwereld (internet,...).

De *firewalls* in Linux zijn bij iedere kernel versie aangepast geweest. Zo hebben we:

- ipfwadm (kernel 2.0)
- ipchains (kernel 2.2)
- iptables (kernel 2.4)

De werking tussen **ipfwadm** en **ipchains** is ongeveer gelijkaardig. Het grootste verschil is de syntax. Om deze reden (en ook door de ouderdom van de 2.0 kernel) gaan we **ipfwadm** niet bespreken in deze documentatie.

In kernel 2.4 kan **ipchains** verder worden gebruikt in plaats van **iptables**. Je moet hiervoor de *ipchains.o* driver laden.

Opbouw IPCHAINS (en IPFWADM)

De basis van **ipchains** is vrij logisch opgebouwd. Men heeft **drie** verschillende *chains*:

- **Input:** binnenkomende pakketten worden hier opgevangen
- **Output:** pakketten die vertrekken vanaf deze computer worden hier verwerkt
- **Forward:** als onze computer een router is kunnen we hierin definiëren wat we willen routen

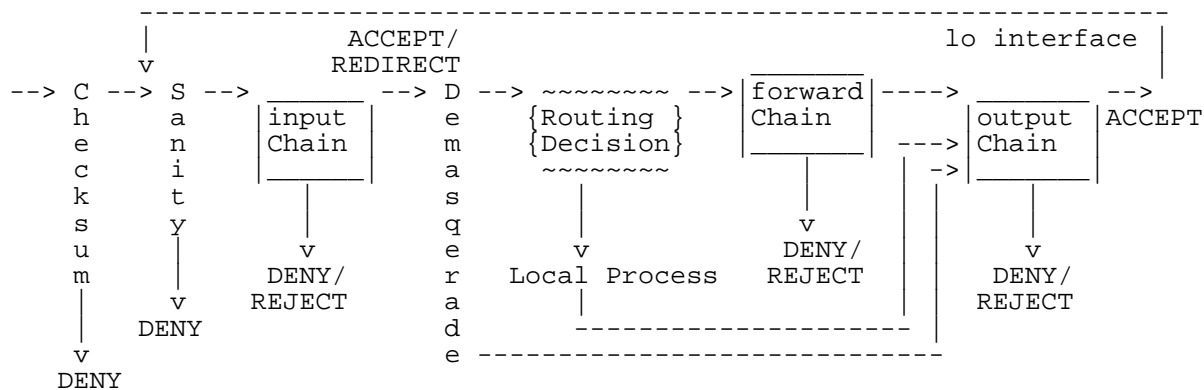
We kunnen ook de *policy* van zo'n *chain* wijzigen. Standaard staat de *policy* op ACCEPT (we laten alles door), maar we kunnen die ook op DENY zetten (alles weigeren). Dit is van belang als we een firewall opbouwen. We kunnen dan kiezen voor "laat alles binnen, behalve x" of "laat niets binnen, behalve x".

We kunnen IP verkeer filteren op *source*, *destination*, *poortnummers*,...

Bij het definiëren van een *rule* (een regel, de definitie van één filter) hebben we de keuze tussen 6 opties wat we met het IP-pakket (dat voldoet aan de definitie van onze filter) gaan aanvangen:

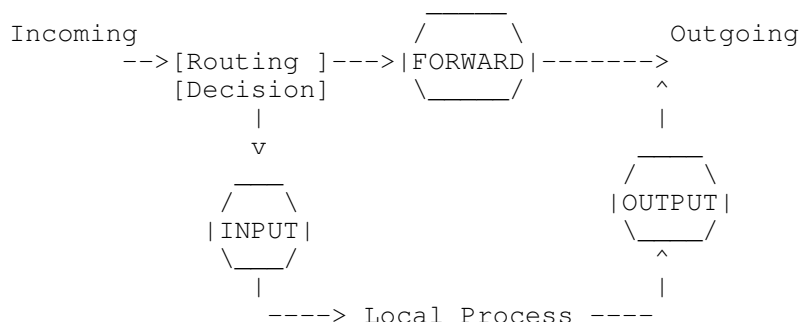
- **ACCEPT**: ontvang de data, zonder filtering (dit wordt vooral gebruikt als de *policy* op DENY staat)
- **DENY**: negeer de data die binnenkomt
- **REJECT**: informeer de *remote* host dat zijn verkeer werd geweigerd
- **MASQ**: voor het gebruik van NAT (Network Adress Translator). Wordt gebruikt bij het routen van verkeer vanaf een intern netwerk ► internet. De *router* gaat de binnengekomen pakketten aanpassen en doorsturen
- **REDIRECT**: doorsturen naar een andere poort/ander adres
- **RETURN**: de *chain*-policy beslist

In volgend diagram wordt werking van de *chains* voorgesteld.



Opbouw IPTABLES

IPTABLES is totaal verschillend van IPCHAINS. Inkomende pakketten die bedoeld zijn voor routing komen niet meer langs de **input** of **output chain** (zie diagram).



De **forward chain** is volledig gescheiden van **input/output**.

Als we verder kijken in het "design" van IPTABLES zien we dat er boven de *chains* ook nog TABLES zijn. Iedere *table* heeft een eigen reeks van *chains*. Er zijn zo 3 standaardtables:

- **filter**: de normale filters met **input**, **output** en **forward chains**
- **nat**: specifiek voor NAT. Bevat de chains **prerouting**, **postrouting** en **output**
- **mangle**: het wijzigen van pakketten. Bevat **prerouting** en **output**

De benamingen van wat we met de pakketten kunnen doen zijn lichtjes gewijzigd:

- **ACCEPT** (zie ipchains)
- **DROP**: tegenhanger van DENY bij ipchains
- **QUEUE**: stuur het pakketje naar "userspace" (waar hopelijk een programma klaarstaat om het op te vangen)
- **RETURN** (zie ipchains)

Je kan ook nog altijd extensies zoals REJECT gebruiken, maar die worden nu gedefinieerd als optie.

Een simpel voorbeeld

Voor de precieze werking van *ipchains* of *iptables* kan je de man-pages raadplegen, maar we raden vooral de HOWTO's aan op <http://www.linuxdoc.org> of <http://netfilter.samba.org>.

Zowel de *ipchains* als *iptables* syntax wordt in dit hoofdstuk beschreven.

Om op te vragen welke *rules* actief zijn kan je volgende commando's gebruiken:

```
ipchains -L
of
iptables -L
```

Om de *policy* van een *chain* te wijzigen (bijvoorbeeld de input *chain*) gebruiken we:

```
ipchains -P input DENY  
of  
iptables -P INPUT DROP
```

Stel dat we één enkele poort willen blokkeren (blokkeren van poort 139 voor inkomend verkeer op een bepaalde ethernetkaart):

```
ipchains -A input -i eth0 -p TCP -d 0/0 139 -j DENY  
of  
iptables -A INPUT -i eth0 -p TCP --destination-port 139 -j DROP
```

Met de optie "-i" definiëren we voor welke netwerkkaart de *rule* geldig is (indien niet aangegeven is de *rule* geldig voor alle kaarten). Met de optie "-d" geef je een adres en poort op, omdat het ip-adres niet relevant is, definiëer je *0/0*, wat *alle* adressen voorstelt (bij *iptables* moeten we geen adres gebruiken). "-j" zegt de netwerklaag wat er met het pakket moet gebeuren indien je een pakket ontvangt dat overeenstemt met de door jou opgegeven criteria. Met "-p" definiëren we het protocol, dit kan bijvoorbeeld ook UDP zijn. We kunnen ook de optie "-s" (source) gebruiken indien je wil filteren op afkomst van een pakket (in dit voorbeeld doen we dit niet).

Voor het invoegen/verwijderen van *rules* kan je volgende opties gebruiken (zowel *ipchains* als *iptables*):

- **A**: append, toevoegen
- **D**: delete, verwijderen
- **R**: replace, wijzigen

Indien je een hele *table* of *chain* wil leegmaken kan je volgende commando's gebruiken:

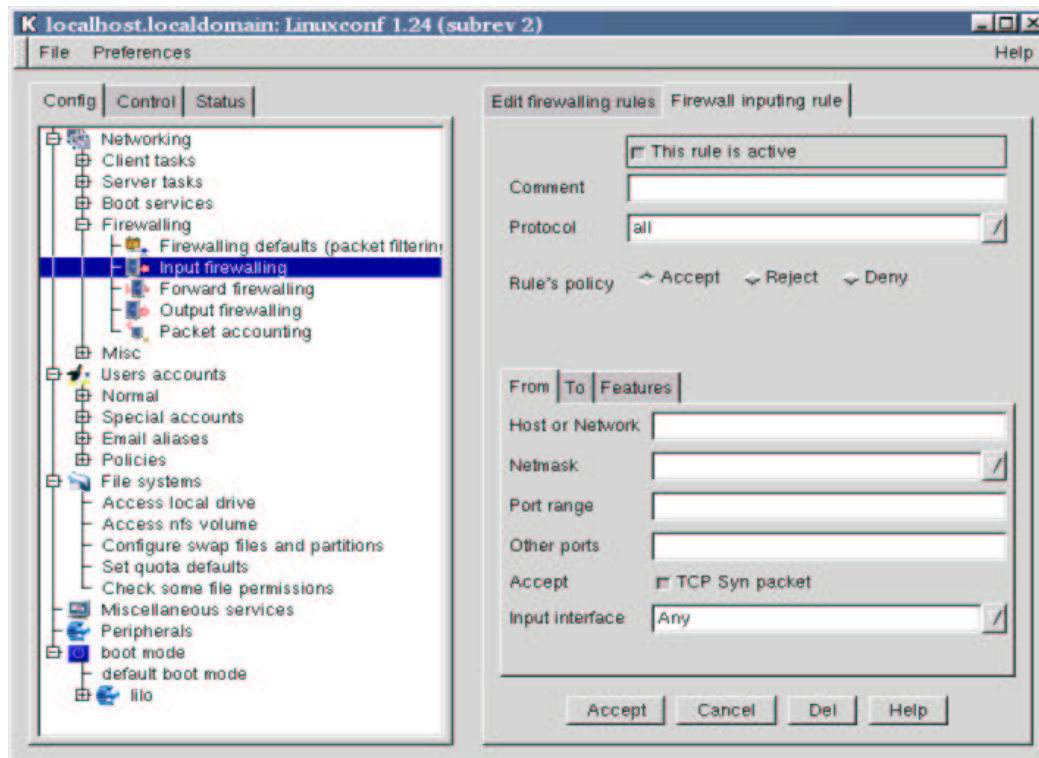
```
ipchains -F of ipchains -F chain  
iptables -F of iptables -F CHAIN
```

GUI tools

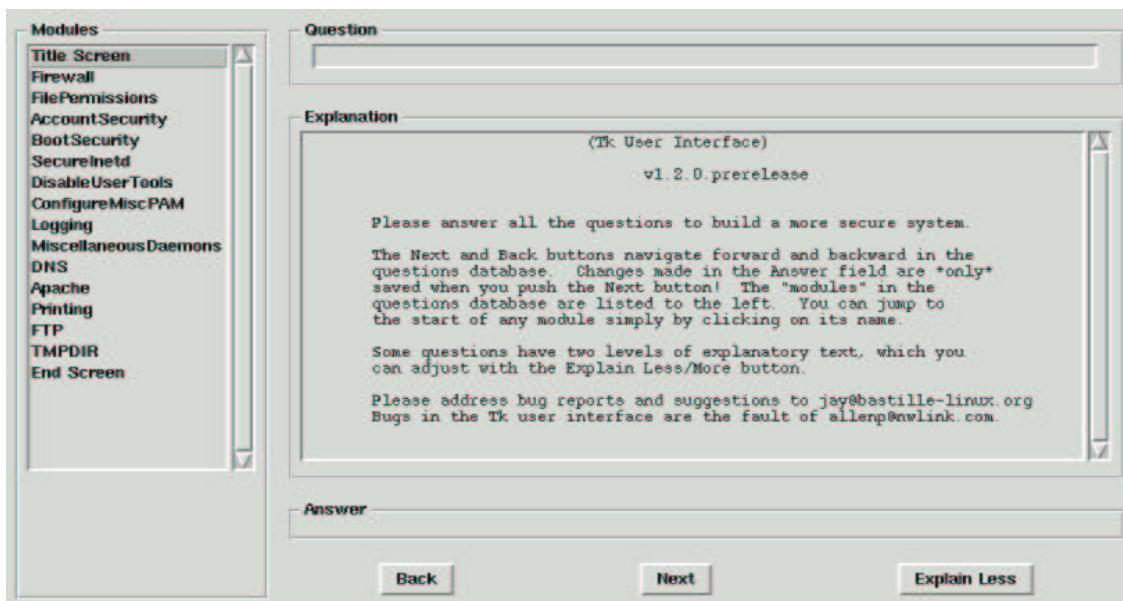
Uiteraard kan je de *firewall* ook wijzigen via GUI tools of automatische scripts.

Je kan onder andere in **Linuxconf** zo'n configuratie vinden. De meeste distributies leveren zelf tools hiervoor (sommige hebben zelfs een configuratie bij de installatie). De meeste tools die je hiervoor vindt kunnen ook *routing* configureren.

De configuratie van de *firewall* vindt je in **Config ► Networking ► Firewalling**

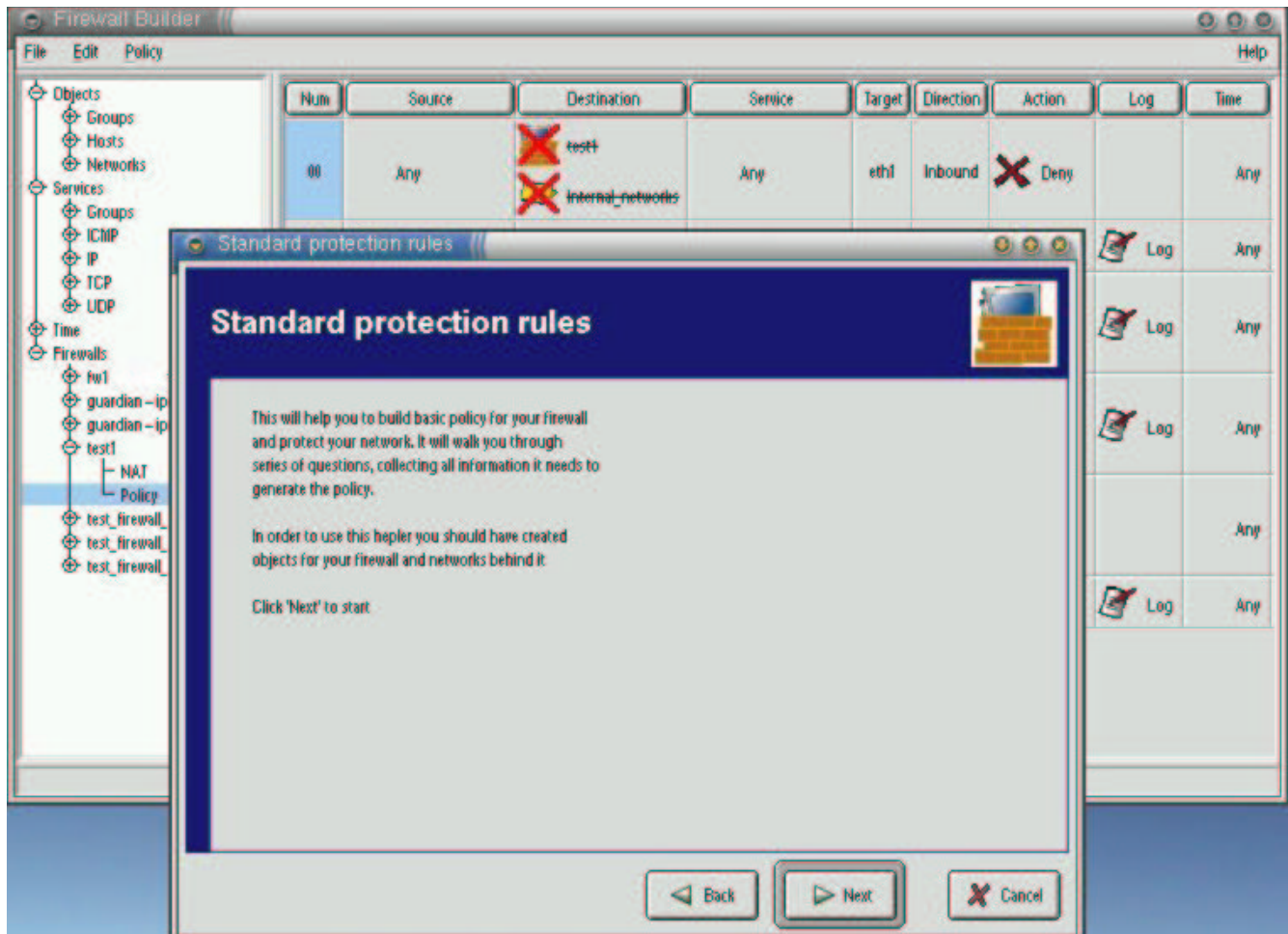


Een aanrader is zeker **Bastille-linux**, te vinden op <http://www.bastille-linux.org>. Dit is een soort *wizard* script die aan de hand van vrij gemakkelijke vragen een firewall opstelt.



Na even vlug opgezocht te hebben op <http://www.freshmeat.net> vind je nog enkele andere voorbeelden zoals:

fwbuilder (te gebruiken met iptables), <http://www.fwbuilder.org>



Het programma is geschreven in C++ en GTK, maar er zijn RPM bestanden verkrijgbaar met de reeds gecompileerde code.

Een alternatief is **Jfwadm**. Dit is een configuratieprogramma voor **ipchains** dat geschreven werd in JAVA.

Zie <http://perso.libertysurf.fr/mickaelf/JFwadmin/>



Firewall Appliance

Velen vinden het misschien handig om hun netwerk aan te sluiten op het internet, maar willen hun netwerk wel afgeschermd en beveiligen van de buitenwereld. Je kan dit door het installeren van een **router/firewall**, maar je kan ook een (oude) PC met Linux ombouwen tot **router/firewall**.

In dit gedeelte is het enkel de bedoeling om te laten zien hoe een basisconfiguratie van zo'n router eruit ziet.

Om een internetlijn te *delen* tussen meerdere gebruikers moet je NAT gebruiken (zie hoofdstuk **IP**). Door NAT kan je de private adressen op jouw netwerk laten vertalen door de router alsof ze van de router zelf komen. Hierdoor wordt het mogelijk om met meerdere computers gebruik te maken van één officieel internet adres.

Je kan de configuratie altijd uitbreiden tot bijvoorbeeld een *proxy-server* die vaak opgevraagde pagina's bewaardt, maar dit is optioneel (op trage computers zet je best geen proxy-server).

De hardware die je nodig hebt:

- 1 PC, 386 of hoger (486 aangeraden)
- 16Mb RAM (met oudere distributies kan men zo laag gaan als 4Mb, maar dit is niet aan te raden).
- 2 netwerkkaarten
- internetconnectie (telenet, ADSL, huurlijn, dial-up,...) en een intern netwerk

Je zet eerst *routing* aan. Je kan dit in **Linuxconf ► Client Tasks ► Routing and Gateways ► Set Defaults ► knop Enable Routing** of de optie "net.ipv4.ip_forward=1" definiëren in */etc/sysctl.conf* (enkel voor RedHat en afgeleide distributies).

Je kan routing ook "at runtime" aanzetten door het commando "echo 1 > /proc/sys/net/ipv4/ip_forward" (bij deze methode gaat de informatie verloren bij een herstart van het systeem, dus hou er rekening mee dat je dit dan nogmaals moet uitvoeren).

Vervolgens moet je bij iedere start van het systeem volgende commando's draaien (we nemen aan dat 192.168.20.0 het netwerkadres van jouw interne netwerk is):

IPCHAINS

```
ipchains -P forward DENY
```

```
ipchains -A forward -s 192.168.20.0/255.255.255.0 -j MASQ
```

Merk op dat we de *policy* van de *forward chain* op DENY zetten. Daarmee gaan we enkel pakketten routen die we definiëren (met andere woorden: het is veiliger zo :-)

IPTABLES

Bij **iptables** moeten we een aparte *table* gebruiken in plaats van de *forward chain*.

```
iptables -P FORWARD DROP
```

```
iptables -A FORWARD -i eth1 -o eth0 -m state --state  
ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
```

```
iptables -t nat -A POSTROUTING -s 192.168.20.0/24 -j  
MASQUERADE
```

eth0 = netwerkkaart aan LAN

eth1 = netwerkkaart aan internet

Cursus Linux

Nota: Dit is enkel een skelet van een configuratie, bijkomende configuratie van de firewall is aangeraden.

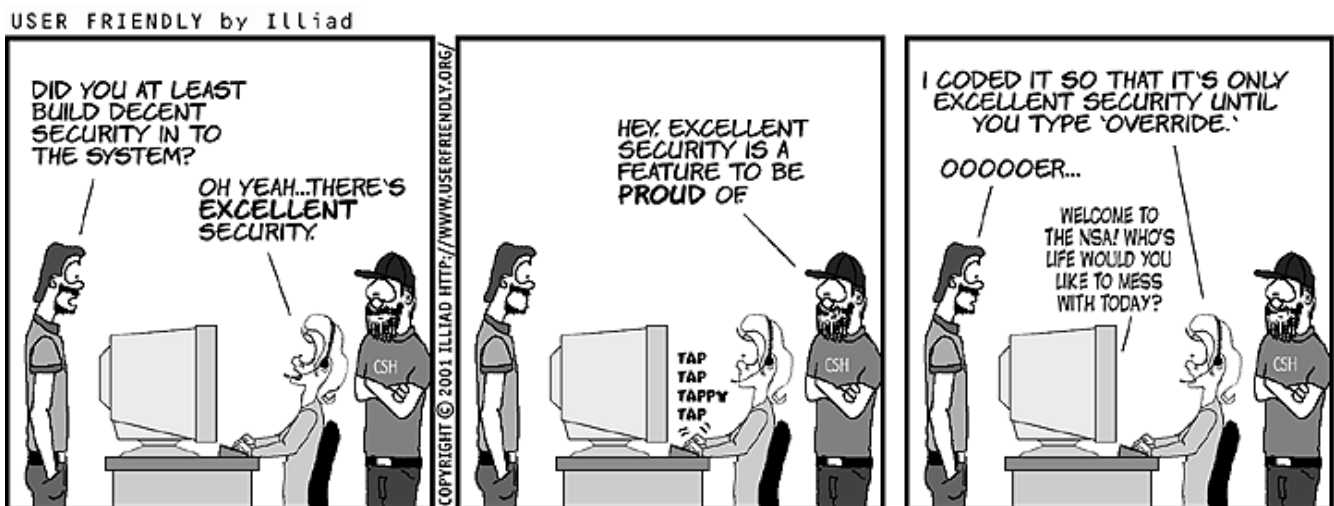
Er zijn ook GUI programma's die dit kunnen configureren (zie eerder in dit hoofdstuk).

Interne PC's

Op de interne PC's moet je het volgende configureren:

- Default gateway = (interne) ip-adres van jouw router/firewall
- DNS = DNS adressen van jouw ISP (of die van een andere geldige DNS server)
- proxy server (enkel nodig bij *Telenet* omdat daar http-verkeer wordt geblokkeerd)

Voor meer gedetailleerde instructies mag je een bezoek brengen aan <http://www.linuxdoc.org> en zoeken naar de *Masquarading HOWTO*.



<http://www.userfriendly.org>

Bestandsdeling *(File Sharing)*



Linux ondersteunt de meest gebruikte *file-sharing* standaarden. Aan de hand van de situatie zal je een aangepast *file-sharing* systeem gebruiken. De meest bekende systemen zijn:

- FTP (Internet file sharing)
- NFS (Unix file sharing)
- SMB (Windows file sharing)

Er zijn nog andere systemen zoals *coda*, maar die worden zelden geïmplementeerd.

FTP

FTP staat voor File Transfer Protocol. Het systeem is enorm efficiënt over een complex netwerk (zoals internet) of lage snelheidsverbindingen (zoals dial-up).

Client

Het gebruik van de FTP-cliënt werd in een eerder hoofdstuk al eens uitgelegd (zie *Internet*).

Server

Er zijn ontelbaar veel ftp-servers beschikbaar voor Linux/Unix, maar de meest gebruikte servers zijn *WU-FTPd* (<http://www.wu-ftp.org/>) en *ProFTPd* (<http://www.proftpd.org/>). Men heeft enkele GUI programma's ter beschikking om deze FTP-servers in te stellen. We gaan er vanuit dat onze server *WU-FTP* geïnstalleerd heeft. We bekijken eerst de configuratiebestanden van *WU-FTP*.

Open het bestand `/etc/ftpaccess`. Hierin vindt je de huidige configuratie van de server. Deze server wordt gestart door **(x)inetd**. Iedere gebruiker op het systeem heeft FTP-toegang met zijn/haar login gegevens.

Om te beletten dat bepaalde gebruikers kunnen inloggen kan je volgende regels gebruiken:

```
deny-uid gebruiker(s)
deny-gid gebruikersgroep(en)
```

Bij de meeste installaties zal je merken dat je op de server kan aanloggen als "anonieme" gebruiker (gebruikers "anonymous" of "ftp" zonder paswoord). Indien je dit wil vermijden moet je de "ftp"-groep en gebruiker uitsluiten.

Cursus Linux

Gebruikers die aanloggen via FTP hebben dezelfde bestandsrechten alsof ze op het systeem zouden aanloggen (ze kunnen enkel schrijven in hun *home* folder). In sommige gevallen kan het handig zijn om bepaalde gebruikers geen toelating te geven om buiten hun *home* folder te *browsen*. Gebruik hiervoor de optie:

```
guestgroup testgroep
```

De gebruikers in "testgroep" zullen dan geen toegang meer hebben tot het hele systeem (enkel hun *home* folder is zichtbaar).

In het configuratiebestand zie je enkele opties en gebruikersprofielen (*guest*, *real* en *anonymous*). *Guest* zijn de gebruikers die we in "guestgroup" hebben geplaatst, *real* betekent "echte gebruikers" en *anonymous* staat voor de onbekende gebruiker.

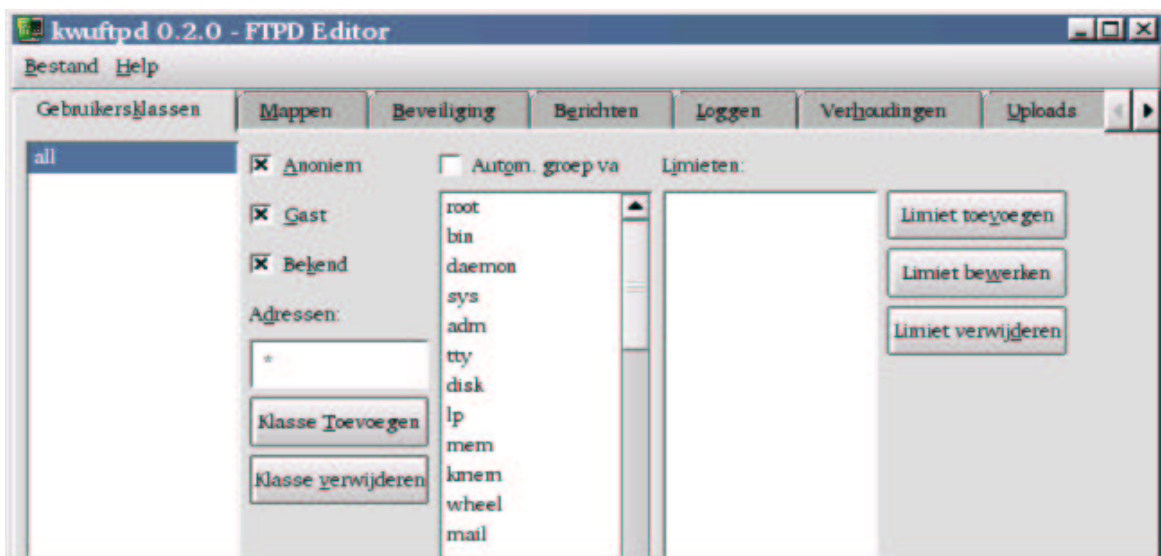
Per profiel kunnen we dan opties aan of uit zetten. Om bijvoorbeeld te voorkomen dat *guest* gebruikers iets kunnen verwijderen gebruiken we de volgende optie:

```
delete no guest
```

Alle transacties die op de FTP-server gebeuren worden opgeslagen in `/var/log/xferlog`.

Verdere informatie vind je met het commando `man ftpaccess`.

Spijtig genoeg is de WU-FTP module in **Linuxconf** niet voldoende om een geavanceerde configuratie te maken, maar gelukkig bestaat er een andere tool: **Kwuftpd**. Je kan deze vinden in het KDE-menu.



NFS

Network File System (NFS) is het file-sharing systeem dat gebruikt wordt door UNIX-machines. Omdat Linux een Unix variant is mag de ondersteuning niet ontbreken.

NFS is een protocol dat efficiënt is op een lokaal netwerk, maar onbruikbaar over internet of trage verbindingen. NFS biedt een speciale *feature*: het is *stateless*. Dit wil zeggen dat indien de verbinding wordt verbroken, de client wacht tot de server weer werkt. Je kan op deze manier bijvoorbeeld de server herstarten zonder dat de verbindingen met de cliënt wordt verbroken (de clients merken het enkel omdat programma's die verbinding willen maken even stoppen). Je kan ook de optie meegeven om deze *feature* uit te schakelen.

Client

Om een server over NFS te bereiken moeten we de **portmap** *service* hebben aanstaan. Indien dit niet het geval is, start deze dan in Linuxconf of met het script `"/etc/init.d/portmap start"`.

Vervolgens het commando om dan een NFS-server aan te koppelen:

```
mount servernaam:/sharedir /plaatsvanmount
```

Om bijvoorbeeld `"/home"` aan te koppelen op de server `"testserver"` aan `/net` geven we het commando:

```
mount testserver:/home /net
```

Bij een oudere versie van *mount* is het nodig om ook de optie `"-t nfs"` mee te geven.

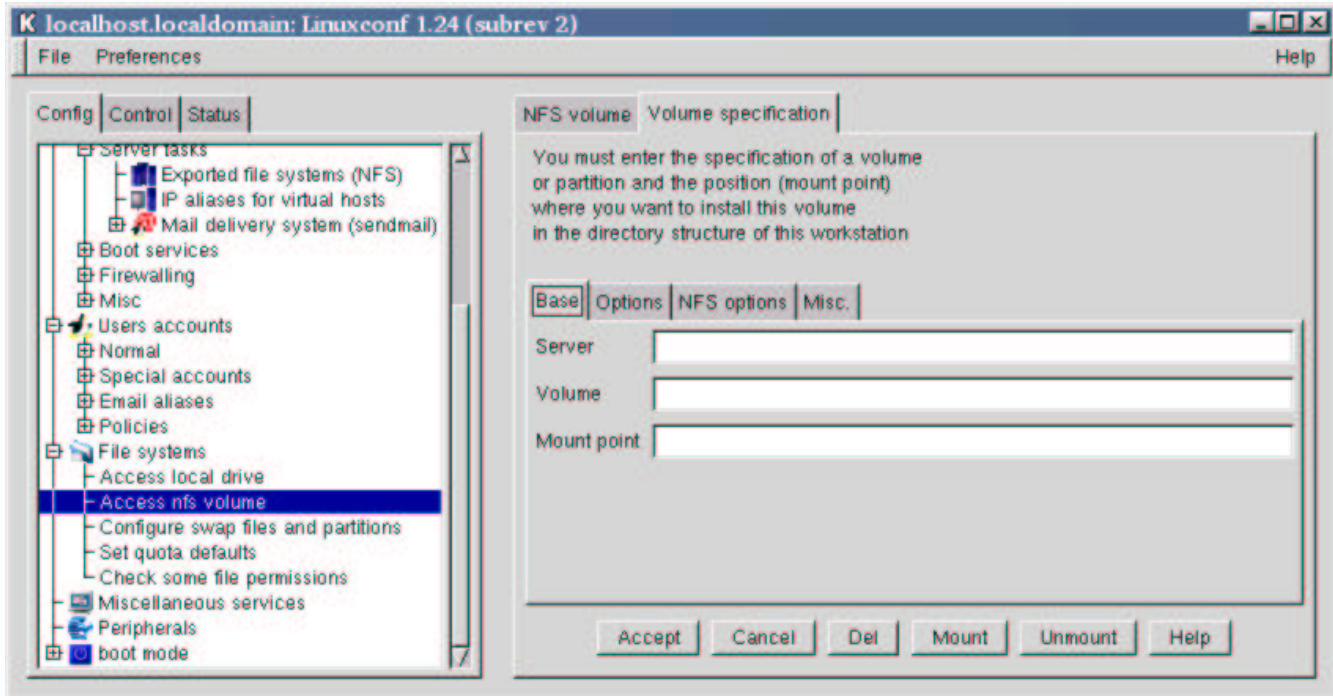
Nota: de servernaam mag ook een IP-adres zijn

Verdere informatie vindt je met `"man mount"`

Om het koppelpunt te *unmounten* (of *af te koppelen*) gebruik je `"umount /plaatsvanmount"`.

Het aankoppelen van een NFS-server kan ook via **Linuxconf**.

Zoek zelf eens uit hoe je een NFS-server kan aankoppelen via dit programma (gebruik eventueel de helpfunctie).



Server

Net zoals bij de client moet je **portmap** starten. Daarnaast heb je **nfs** en (liefst ook) **nfslock** nodig. Je start deze daemons met:

```
/etc/init.d/nfs start  
/etc/init.d/nfslock start
```

Als je de NFS-server vaak gebruikt, start deze dan automatisch via Linuxconf.

De server gaat het bestand `"/etc/exports"` gebruiken om te controleren welke *shares* beschikbaar zijn en wie erop mag inloggen.

Een share is aangegeven met de volgende syntax:

```
directory machine1(option11,option12) machine2...
```

- directory: de directory die je deelt (dit is ook de volumenaam)
- machine: IP-adres, netwerkadres of hostnaam van de computer
- option: opties zoals **ro** (read only) of **rw** (read write)

Een voorbeeld:

```
/usr/local 192.168.0.1(ro) 192.168.0.2(rw)
```

Hiermee delen we `"/usr/local"`. Er zijn maar twee machines die kunnen inloggen, waarvan er slechts één kan schrijven.

Indien je geen *machinenaam* meegeeft, dan heeft iedere client access tot de share.

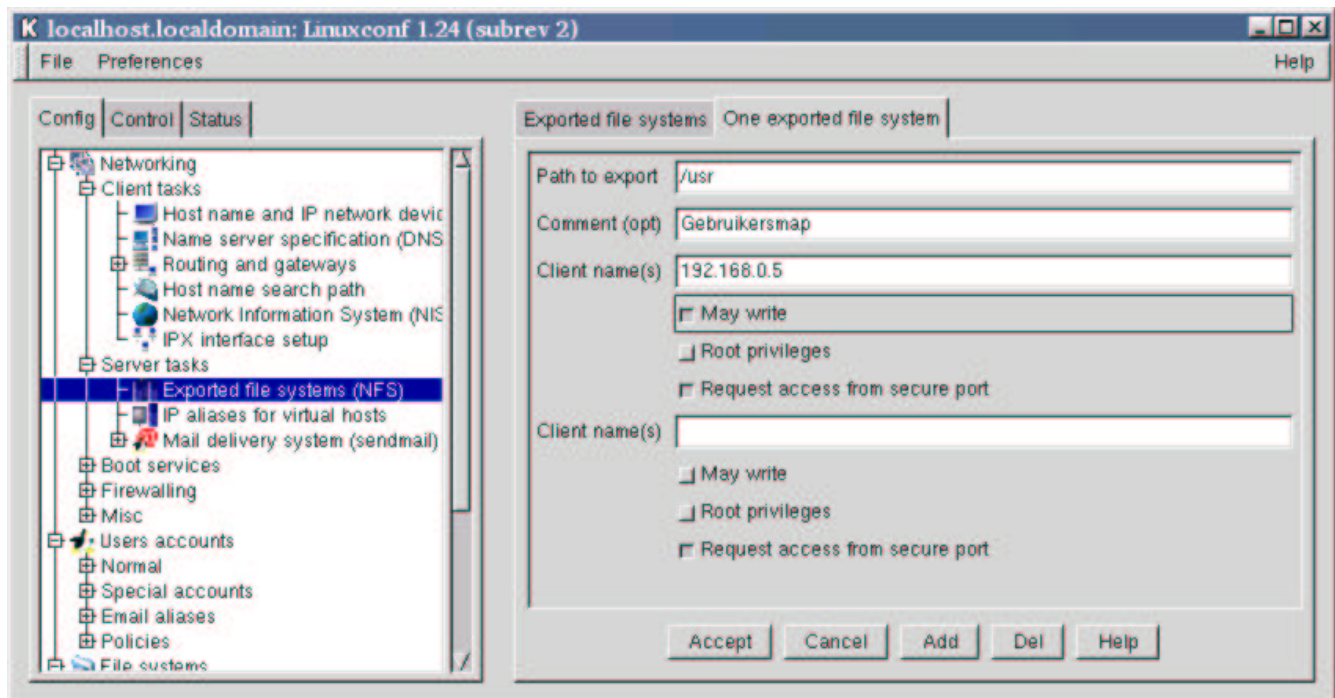
Cursus Linux

Om wijzigingen in `/etc/exports` actief te maken kan je de NFS server herstarten of het commando `exportfs -ra` starten aan de command-line.

Je kan ook gebruikers *mappen*. Zonder NIS (UNIX domein) is dit systeem zo lek als een zeef omdat iedereen in principe "root" kan zijn op een willekeurige PC. Met NIS kan je "netgroepen" gebruiken die wel veilig zijn.

Voor verdere informatie consulteer je "man exports".

Voor de liefhebbers van GUI-tools: je kan de *exports* ook instellen via **Linuxconf**.



SMB

SMB staat voor **Server Message Block**. Het is een systeem dat gebruikt wordt om tussen *Windows* machines bestanden te delen. Je kan op een Linux machine ook toegang verkrijgen tot een Windows-netwerk via *Samba*. De mogelijkheden zijn onbeperkt, je kan zelfs een *Primary Domain Controller* emuleren. Meer informatie vind je op <http://www.samba.org>.

Client

Je kan toegang verkrijgen tot een *remote* Windows PC (of Samba Linux server) met het mount commando. Indien je op een NT/2000 computer aanlogt moet je een gebruikersnaam ingeven:

```
mount -t smbfs //servernaam/sharenaam /koppelpunt
```

Als je op een NT-server aanloggen:

```
mount -t smbfs -o username=gebruiker //servernaam/sharenaam /koppelpunt
```

Je krijgt bij deze twee commando's telkens een prompt om het paswoord in te typen. Je kan dit automatiseren door in het commando een extensie op de "-o" optie te gebruiken:

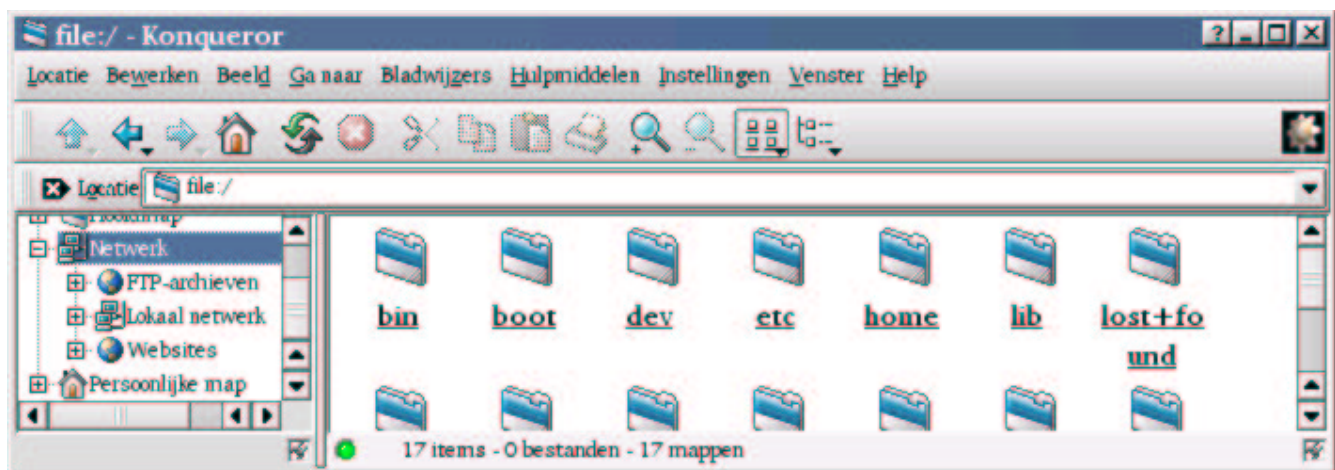
```
mount -t smbfs -o username=gebruiker,password=paswoord //servernaam/sharenaam /koppelpunt
```

Nota: Merk op dat we de servernaam en share definiëren met "/"(slash) tekens. Dit omdat de "\"(backslash) een speciale betekenis heeft in de shell.

- **Windowsmanier:** \\windowsserver\sharenaam
- **Linux/Unix:** //windowsserver/sharenaam
- **Linux/Unix (alternatief):** \\\\windowsserver\\sharenaam

Je kan meer informatie verkrijgen via "man smbmount". Kijk ook eens naar het programma "smbclient" (zie ook "man smbclient").

Modernere Linuxdistributies kunnen Windows-shares toegankelijk maken via bijvoorbeeld de verkenner van KDE.



Server

Het instellen van een **Samba** server is vrij complex. Je gaat eerst de opties in `"/etc/samba/smb.conf"` bekijken.

Globale definitie:

```
[global]
```

Instellingen die gelden voor de volledige server

```
workgroup = groepnaam
```

Naam van de werkgroep. Als je deel wil uitmaken van een domein moet je hier de domainnaam invullen.

```
netbios name = servernaam
```

De *Windowsnaam* die je aan jouw Linuxcomputer wil geven.

```
security = share
```

Deze optie is van belang bij het authenticeren van *remote* gebruikers. Met *share* lijkt dit op de sharing in Windows 9x (we laten iedereen toe om te schrijven of lezen). Als je gebruikersnamen en paswoorden wil controleren moet je deze optie op *user* zetten. Als je paswoorden wil controleren aan de hand van een andere server gebruik je de optie *domain* (authenticeren via het domein) of *server*.

```
password server = <NT-Server-Name>
```

Enkel nodig indien "*security=server*". Indien "*security=domain*", vul dan een "*" in als servernaam.

```
encrypt passwords = yes
```

Als er geen Windows95 PC's in het netwerk staan, zet dit op "yes".

```
domain master = no
```

Doe mee aan de "master browser elections"

```
os level = 33
```

Hoe hoger dit getal, hoe hoger de kans is dat deze Linuxcomputer *master browser* wordt.

```
preferred master = no
```

Verhoogt de kans dat onze server *master browser* wordt.

Opgelet: zet "*domain master*" uit indien onze computer geen *Primary Domain Controller (PDC)* is en er wel een domein aanwezig is op het netwerk met een NT/2000 server. Een Windows PDC moet volgens zijn software ook *master browser* zijn. Indien de *Samba* server toch de "master browser elections" wint kunnen we de software op de Windows server in de war brengen.

```
domain logons = no
```

Start een eigen domein waarvan deze computer de Primary Domain Controller is.

Share definities:

We beginnen altijd met de "sharenaam" tussen vierkante haken

```
[public]
  path = /usr/somewhere/else/public
  public = yes
  only guest = yes
  writable = yes
```

Dit is een voorbeeld van een share die voor iedereen toegankelijk is (met *security=share*).

```
[myshare]
  comment = Mary's and Fred's stuff
  path = /usr/somewhere/shared
  valid users = mary fred
  public = no
  writable = no
  write list = mary
```

Hier definiëren we welke gebruikers toegang hebben. "Mary" kan erop schrijven, "Fred" kan enkel lezen (dit kunnen domein of locale gebruikers zijn).

Je kan in plaats van een gebruiker ook een groep definiëren (gebruik dan "@*groepnaam*").

Als je gebruikers wil laten authenticeren op jouw server moet je rekening houden dat:

- De gebruikers zijn aangemaakt op het systeem
- De gebruikers zijn aangemaakt in Samba

Het aanmaken van de gebruikers op het systeem kan via **Linuxconf** (zie eerder).

Daarna moet je Samba deze gebruiker laten "herkennen". Doe dit met het commando:

```
smbpasswd -a gebruikersnaam
```

Geef deze gebruiker dan een paswoord dat geldig zal zijn vanaf *Windows* PC's. Indien gewenst kan je "unix password sync" aanzetten in "smb.conf" om het Unix paswoord mee te wijzigen wanneer het **Samba** paswoord wijzigt.

Als je gebruikers wenst aan te maken voor de **Sambaserver**, maar je wil niet dat ze gebruik maken van de Unix-programma's en logins, dan kan je de aangemaakte systeemgebruikers "locken" (blokkeren). Geblokkeerde accounts blijven werken met **Samba**. Als je **Samba** accounts wil blokkeren, doe dit dan met "smbpasswd".

Om jouw server in een domein te plaatsen (en de gebruikers daarop wil laten authenticeren) heb je twee opties:

- Gesimuleerde domeinen
- Winbind

Door middel van **Winbind** (enkel te vinden in de recentere uitgaves van **Samba**) kan je de server laten deel uitmaken van een Windowsdomein. Als een gebruiker aanloft wordt hij/zij geauthenticeerd door de *Windows Domain Controller*. Als het gaat om een geldige gebruiker, dan wordt er automatisch een tijdelijke gebruiker aangemaakt die rechten heeft op het systeem.

De server integreert zich dan compleet in het Windowsdomein.

In de *gesimuleerde omgeving* moet je eerst locale gebruikers aanmaken die overeenstemmen met die in het domein. Je maakt dan op de PDC een *computeraccount* aan voor onze server. Dan typ je volgend commando op de Linuxserver:

```
smbpasswd -j DOM -r DOMPDC
```

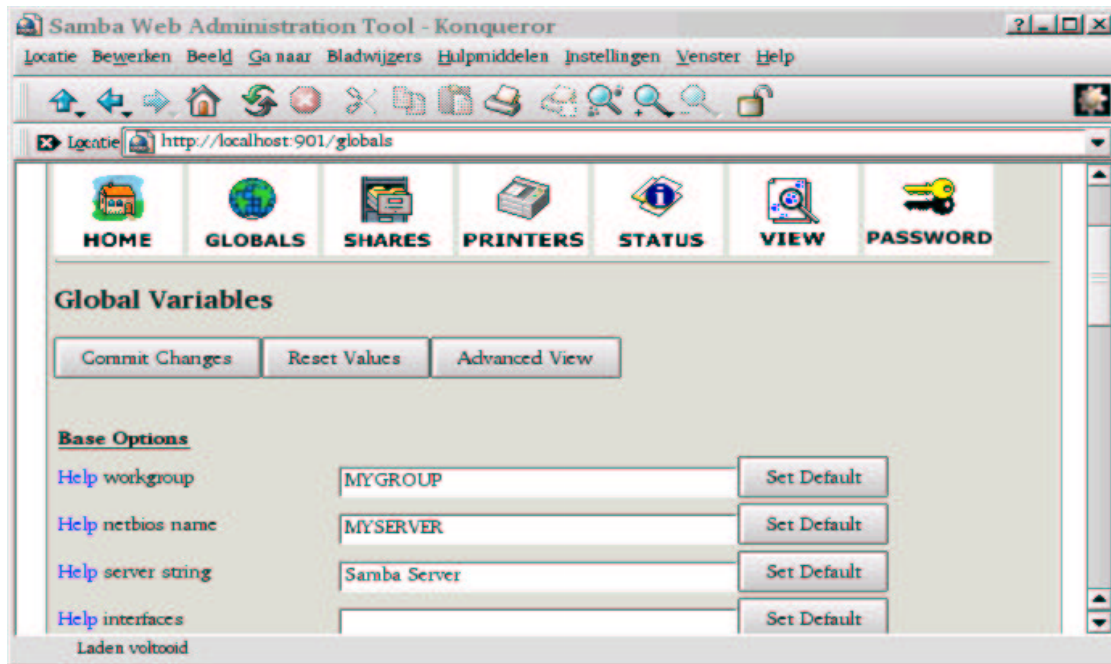
"Dom" is de naam van het domein, "DOMPDC" de naam van de PDC. Zet in "smb.conf" de optie "security=domain". Het maakt niet meer uit welke paswoorden er in de **Samba** configuratie staat, authenticatie gebeurt nu via de PDC (enkel paswoord!). Dit systeem heeft een enorm nadeel: je moet nog altijd de gebruikers lokaal aanmaken.

Ook in dit geval zijn er GUI-tools die je kunnen helpen. Een volledige lijst kan je vinden op <http://www.samba.org/samba/GUI/>.

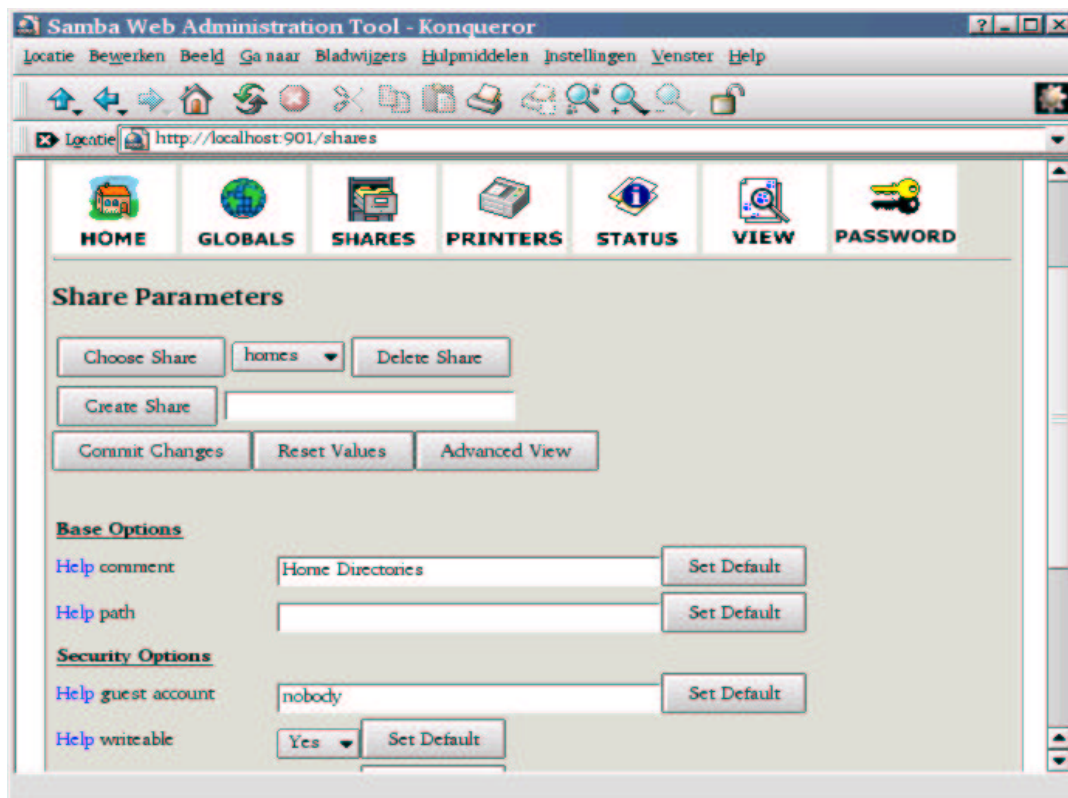
De handigste GUI-tool wordt meegeleverd met **Samba** en luistert naar de naam **SWAT**. Je kan de tool gebruiken door met een *internet browser* naar het adres <http://localhost:901> te surfen.

Controleer wel of **SWAT** geactiveerd is in (x)inetd.

Cursus Linux

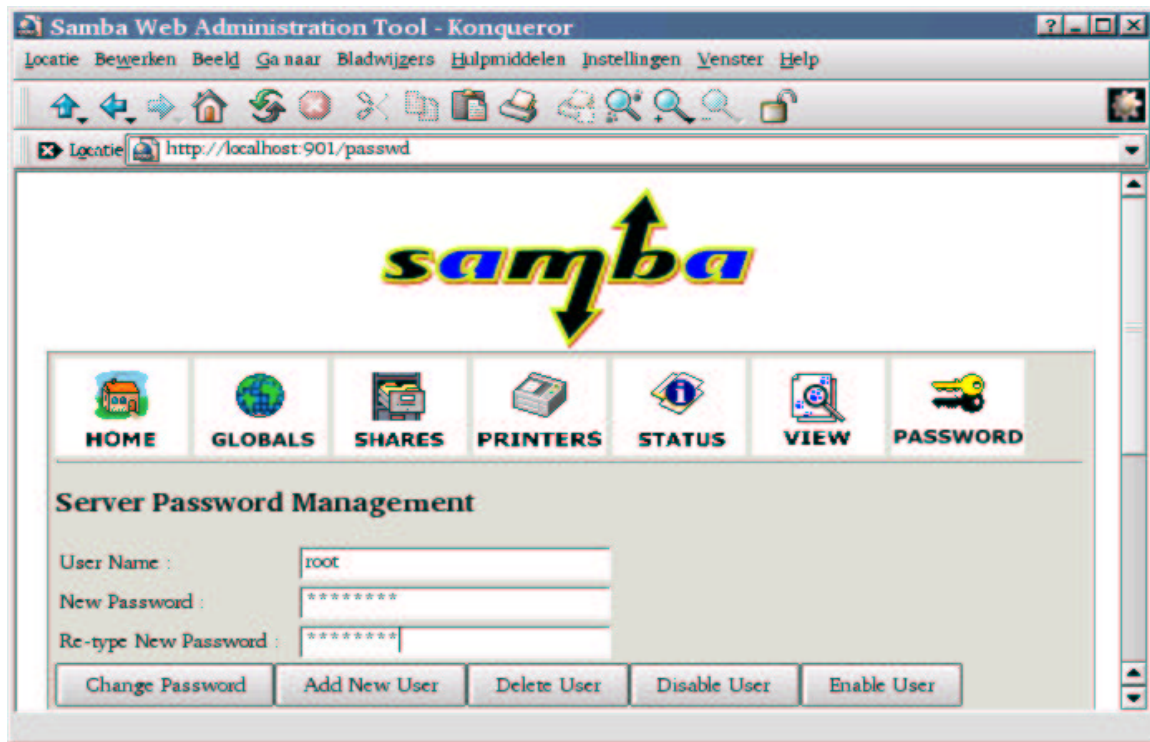


Globale instellingen



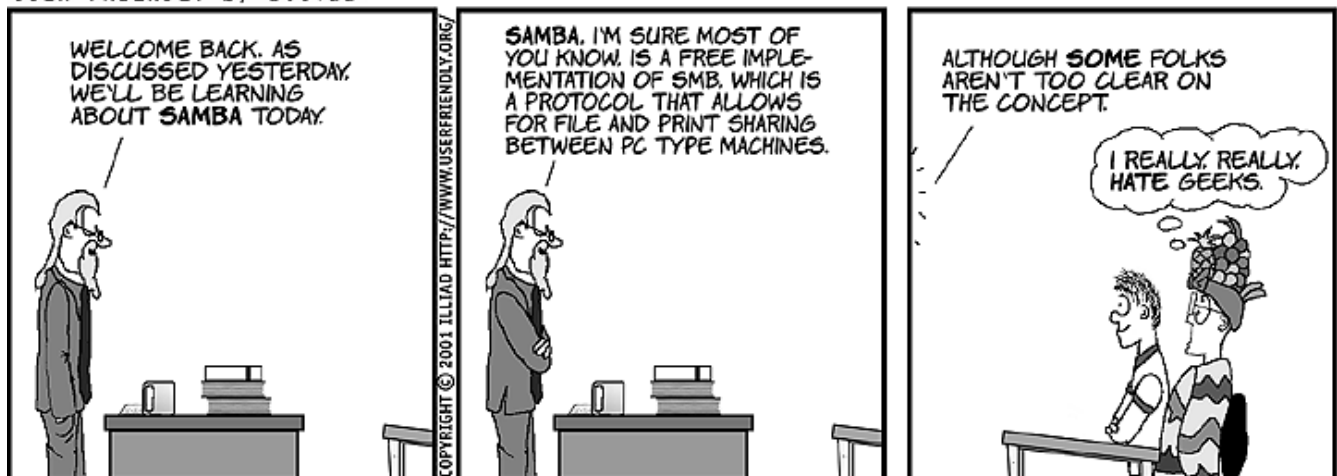
Aanmaken, verwijderen en aanpassen van *shares*

Cursus Linux



Gebruikersbeheer in Samba

USER FRIENDLY by Illiad



<http://www.userfriendly.org>

NIS

NIS staat voor *Network Information System*. Met dit systeem kan men gebruikers bekend maken over een heel netwerk. Je kan NIS vergelijken met een *domain* onder Windows.

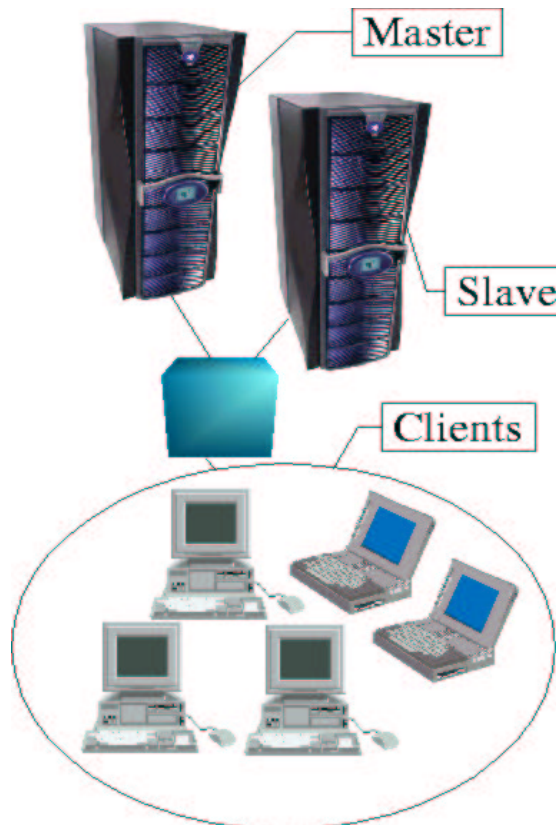


NIS werd ontwikkeld door *Sun Microsystems*, en was voordien ook gekend onder de naam **Yellow Pages (YP)**.

Een NIS zal je gebruiken wanneer je op verschillende Linux/Unix machines wil werken met dezelfde *users*. Gebruikers en paswoorden worden dan centraal bijgehouden.

Werking

Je kan op een netwerk één of meerdere NIS-servers plaatsen. Je kan meerdere NIS-servers hebben die verschillende domeinen beheren, of je kan ook over *slave* servers beschikken. Deze servers worden dan gebruikt als backup systeem.



De NIS server

Eerst moet je de **portmapper** *service* starten (indien dit niet het geval was).

```
/etc/init.d/portmap start
```

Geef jouw server een geldige *NIS-domainname*. Indien je met het commando "domainname" een "(none)" terugkrijgt, dan is dit nog niet ingesteld. Stel het in met het commando "domainname *naamdomein*".

Je kan de bestanden `/var/yp/securenets` en `/etc/ypserv.conf` wijzigen indien gewenst (zie "man `ypserv.conf`"). In de meeste gevallen is dit niet nodig.

Start nu de **ypserv** *service* met het initscript in "`/etc/init.d/ypserv start`".

Controleer of de server werkt met het commando "`rpcinfo -u localhost ypserv`". Je zou een uitvoer moeten krijgen in de aard van:

```
program 100004 version 1 ready and waiting
program 100004 version 2 ready and waiting
```

Om de allereerste keer de server te gebruiken moet je het programma "`/usr/lib/yp/ypinit -m`" draaien. Volg dan de instructies op het scherm.



Vanaf dit punt draait de server. Als je een wijziging hebt gemaakt aan de gebruikersgegevens op de server, dan moet je deze gegevens doorsturen naar de server. Dit kan door het "make" commando te gebruiken in de "`/var/yp`" map.

Het is aangeraden ook nog de "`yppasswdd`" *service* te draaien zodat gebruikers hun paswoord vanop afstand kunnen wijzigen.

Als alles werkt, configureer de server zodat de juiste services worden opgestart bij een *reboot*.

De NIS cliënt

Start de **portmapper** *service* indien dit nog niet het geval zou zijn.

Het installeren van de cliënt kan via Linuxconf of een tekstbestand.

Plaats in "`/etc/yp.conf`" een regel in de vorm van:

```
ypserver x.x.x.x
```

(waarbij `x.x.x.x` het ip-adres is van de server, zie "man `yp.conf`" voor verdere info, je kan in dit bestand meerdere servers plaatsen).

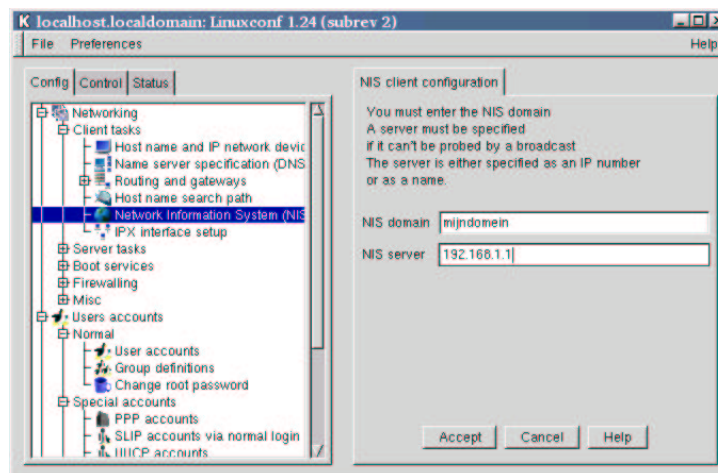
Start de "`ypbind`" *service*.

Cursus Linux

Vanaf dit punt zou je gebruik moeten kunnen maken van "users" die op de server gedefinieerd zijn.

Als het niet werkt, bekijk dan eens `/etc/nsswitch.conf`. Hierin wordt de volgorde van *resolving* (zowel gebruikers, groepen en hosts) weergegeven. Zorg ervoor dat er "nis" achter de items *shadow*, *users*, *group* en *hosts* staat.

Er zijn ook GUI mogelijkheden. Bij de installatie van de meeste Linuxdistributies heb je die optie... Of je kan ook eens een kijkje gaan nemen in Linuxconf.



USER FRIENDLY by Illiad



<http://www.userfriendly.org>

Virtuele computers

Linux biedt de mogelijkheid om meerdere "Operating Systems" naast mekaar te draaien. Dit is trouwens ook mogelijk met andere OS'sen gebaseerd op het "virtual memory model" zoals Windows NT/2000 en UNIX, maar Linux blijkt in deze opstelling beduidend efficiënter te zijn dan zijn tegenhangers. Het is bijvoorbeeld niet uitgesloten dat Windows in een Linuxvenster sneller werkt dan dat je Windows "native" zou installeren op een PC.



Een *virtuele computer* is een emulatie van een volledige PC-omgeving (BIOS, geheugen, schijfruimte,...). Bij de meeste implementaties ben je wel gebonden aan de processorarchitectuur (bijvoorbeeld: op een X86 PC kan je enkel virtuele computers met X86-processor emuleren).

Er zijn verschillende producten op de markt:

- **VMWare:** De meest gekende (commerciële) producent van dit soort producten. Onder hun Linuxversie kan je DOS, Windows 3.1/9x/NT, Linux, *BSD draaien. Ze hebben ook een product voor Windows NT (en hoger) waardoor je onder Windows een virtuele Linux kan draaien. <http://www.vmware.com/>
- **Win4Lin:** Net zoals VMWare is dit een commercieel product. De werking is ongeveer gelijkaardig, maar er is geen Windowsversie. <http://www.win4lin.com/>
- **Plex86:** Een open source tegenhanger van VMWare. Dit programma is niet echt gebruiksvriendelijk om te installeren, maar het werkt... <http://www.plex86.org/>
- **Bochs:** Net zoals Plex86 is dit Open Source. Het voordeel van dit programma is dat je een X86 computer kan emuleren op een niet-X86 computers. Je kan bijvoorbeeld Windows draaien op een Linux PPC(PowerMac). Er zijn evenwel nadelen verbonden aan dit systeem omdat het trager werkt. <http://bochs.sourceforge.net/>

De "straffe manne" van *IBM* gaan nog een stap verder. Zij verkopen *mainframes* (grote machines met honderden processors en Terabytes geheugen) met virtuele Linuxpartities. Zo'n mainframe kan gemakkelijk enkele duizenden virtuele Linux-servers opstarten. Op die manier verleggen de techniekers van *IBM* de mainframemarkt van de financiële wereld (voor grote databases, waarvoor vroeger een mainframe diende) naar Internet Service Providers (ISP) die heel hun computerpark met één mainframe kunnen vervangen.

Voor een ISP geeft dit volgende voordelen:

- minder ruimtevoorziening (inclusief koeling)
- minder elektriciteitsverbruik
- minder problemen met bekabeling
- flexibiliteit (computerkracht kan naar servers gevoerd worden die het écht nodig hebben)
- snelle aanmaak van nieuwe servers (volgens de reclame van IBM in minder dan 5 minuten)
- snelle communicatie tussen servers (de netwerksnelheid is de snelheid van het geheugen, welke enkele GB/sec bedragen)
- enorm stabiel (mainframes gaan zelden stuk, en zelfs als dit gebeurt merkt niemand er iets van behalve de *IBM* technieker)



Eén nadeel is natuurlijk wel het prijskaartje (...maar 1000 PC's zijn ook niet goedkoop)

Meer info op <http://www-1.ibm.com/servers/eserver/zseries/os/linux/>

Praktijk

We gaan zelf eens een virtueel systeem installeren.

Probeer eerst de **VMWare** software te bemachtigen. Er is een "trial" versie beschikbaar die we kunnen gebruiken. Wees er zeker van dat je de RPM-versie download. Hou ook een Windows-installatie CD bij de hand.

Na de installatie met het "`rpm -Uvh naambestand.rpm`" commando moet je de configuratie starten. Dit is een *shellscript* dat enkele vragen zal stellen hoe jouw virtuele computer moet geïnstalleerd worden.

Het configuratieprogramma start je met "`/usr/bin/vmware-config.pl`".

Enkele voorbeelden van vragen:

- netwerkadressering
- dhcp-server voor virtuele computers
- file sharing (langs de *host* kant)

Na de configuratie worden de nodige drivers geladen. Je kan dan (in een GUI venster) **VMWare** starten met het "vmware" commando. Je krijgt dan een venster te zien met allerlei menu's en opties. In "**File ► New**" (of via wizard) kan je een nieuwe computer aanmaken.

Je kiest dan hoe groot de harde schijf en het geheugen moeten zijn voor jouw virtueel systeem (neem nooit waarden groter dan het *host* systeem!).

De harde schijf van het virtuele systeem zal worden opgeslagen in een bestand (het is ook mogelijk een bestaande partitie te gebruiken...).

Je hebt de keuze tussen drie verschillende netwerkvormen: **bridge** en **host-only**. In **bridge** is de virtuele computer lid van het netwerk waar de *host* computer aanhangt. Bij **host-only** krijg je een ip-adres van de *host* en maak je geen deel uit van het externe netwerk.

Als je dan het systeem voor de eerste keer opstart zie je een BIOS scherm (dit zal iedere keer tevoorschijn komen als je de virtuele computer opstart). Je kan dan net zoals bij een gewone computer Windows of een ander OS installeren (hint: het is nu het moment om jou Windows CD te gebruiken).

Windows zal een AMD-netwerkadapter detecteren (dit is een virtuele netwerkadapter waarmee je kan communiceren met de *host*). Om extra features te installeren zoals *full-screen operation* en *mouse-switching* kan je de *virtuele* driver diskette aankoppelen (**Settings ► VMWare Tools install**) en de drivers installeren in *Windows* (alsof het een diskette is).

Je kan ook de virtuele computer in full-screen mode draaien zodat je tussen schermen kan *switchen* met de **CTRL-ALT-Fx** toetsen.

Indien de *virtuele computer* blokkeert door een softwarefout, zal dit geen gevolgen hebben voor de *hostcomputer*. Tot op heden is het niet mogelijk om DirectX spelletjes te spelen onder VMWare.

In de volgende screenshot zie je het opstartscherm van *Windows 98*.



Cursus Linux



Hou er rekening mee dat je nog altijd een geldige *Windows* licentie moet hebben, ook al draait deze onder een Virtuele Machine.

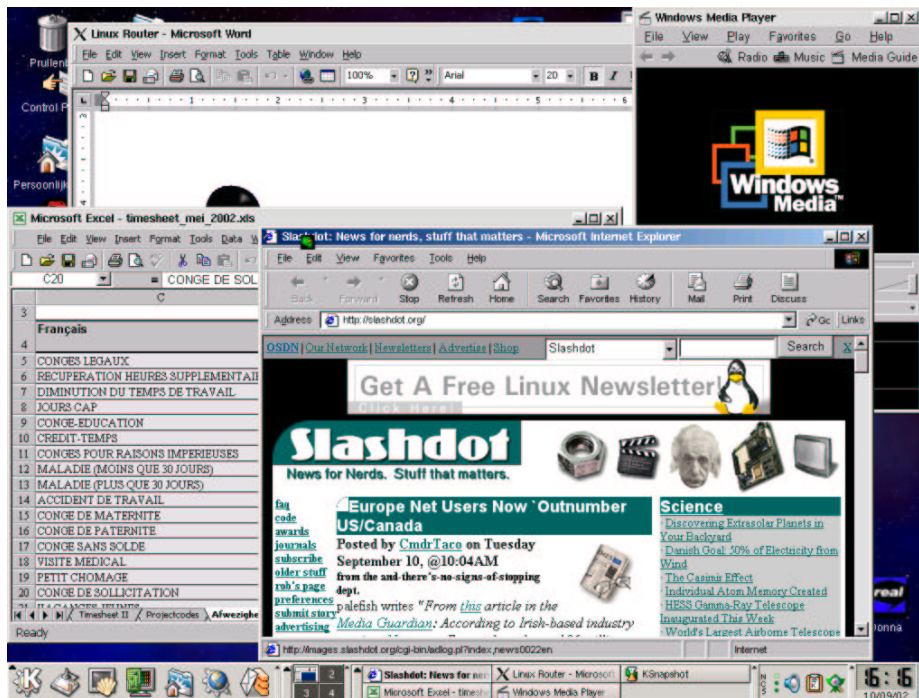
Met WINE heb je geen *Windows* licentie meer nodig (zie hoofdstuk WINE). Je hebt nog wel licenties nodig voor de software die erop draait.

WINE

Wine is een *Windows emulatie laag*. Ironisch genoeg betekent WINE "*Wine Is Not an Emulator*". Met **Wine** is het mogelijk om *Windows executables* (programma's) te draaien onder *Linux*.

Momenteel zijn er 3 organisaties die een versie van **Wine** uitbrengen:

- **WineHQ**: Een Open Source implementatie van de Wine. Deze versie is de basis van alle andere Wine versies. In de meeste distributies levert men deze versie mee, maar je kan Wine downloaden op <http://www.winehq.org>. De mogelijkheden zijn niet spectaculair. Simpele programma's zoals *paint* en *notepad* zullen wel werken, maar ingewikkelde programma's werken meestal zeer slecht.
- **Codeweavers Wine**: Commercieel ondersteunde versie van Wine. De programmeurs van Codeweavers verkopen twee versies: *Crossover Plugin* en *Crossover Office*. *Crossover Plugin* is gemaakt om browser plugins voor *Windows* onder *Linux* te gebruiken. Hiermee wordt het mogelijk om *RealPlayer*, *Quicktime* of de aller nieuwste versie van *Flash* te gebruiken. *Crossover Office* is bedoeld om applicaties zoals *Microsoft Office*, *Lotus Notes* of *Quicken* te draaien. Beide producten werken perfect met de vernoemde applicaties. <http://www.codeweavers.com>
- **Transgaming WineX**: Net zoals bij Codeweavers is dit een commercieel product. Het doel van deze versie is compleet verschillend. Het is niet de bedoeling om gewone programma's op *Linux* te draaien, maar wel *Windows DirectX* spelletjes. De spelletjes die voor *Transgaming* een prioriteit zijn werken perfect (zoals oa. *Warcraft III* en *GTA3*). Andere games werken niet zonder problemen. Spijtig genoeg is het aantal ondersteunde games beperkt. <http://www.transgaming.com>



Kernel compile

Je kan een kernel zelf compileren. Op die manier heb je een kernel die perfect aan jouw wensen voldoet. Zo kan je bijvoorbeeld kernels compileren met minder "features", maar die sneller zijn. Een bijkomende reden om een nieuwe kernel te compileren is dat er in de nieuwe versie misschien drivers zitten die je nodig hebt.



In de meeste gevallen is het echter niet nodig om een kernel te compileren. De distributies leveren een kernel met een volledige collectie aan "features" zodat hercompileren voor een bepaalde driver onnodig is. Ook kernel *upgrades* kunnen uitgevoerd worden via een RPM pakket (lees wel aandachtig de installatieprocedure!).

Waar

Je kan de laatste versie van de kernel downloaden op <http://www.kernel.org>, de meeste distributies stellen ook de source ter beschikking van de laatste geteste kernel in RPM vorm (bij Redhat heet deze RPM *kernel-source*).

Pak daarna het pakket uit in de `/usr/src` map (de broncode komt in `/usr/src/linux` of een variant hiervan terecht), of installeer de RPM.

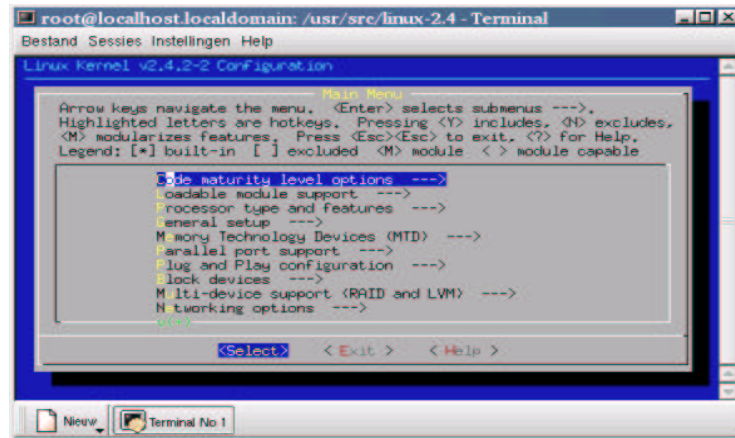
Compileren

Ga naar de map (`/usr/src/linux`).

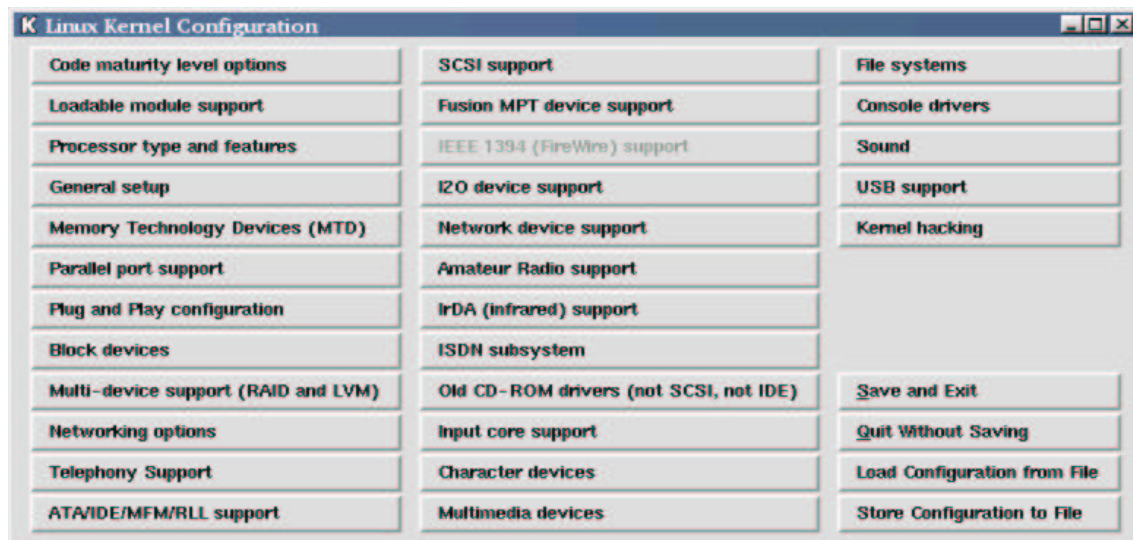
Eerst moeten we de configuratie van de nieuwe kernel klaarmaken. Dit doen we door het commando "make config". Je krijgt dan een lijst van vragen waarop je moet antwoorden (aan de hand van deze antwoorden wordt een *custom* kernel gemaakt die aan jouw wensen voldoet).

Om de configuratie gebruiksvriendelijker te maken kan je ook gebruik maken van "make menuconfig" of "make xconfig".

Cursus Linux



menuconfig



xconfig

Met "save&exit" kan je het configuratieprogramma afsluiten.

Gebruik nu het commando "make dep". Dit dient om de afhankelijkheden te verwerken.

Het commando "make modules" gaat het modulaire gedeelte van de kernel compileren (meestal zijn dit drivers).

Daarna moet je de *modules* installeren op schijf (meestal komen deze in */lib/modules/kernel-versie* terecht).

Geef het commando "make modules_install".

Cursus Linux

Nota: Als je een kernel aan het compileren bent die exact hetzelfde versienummer heeft dan een al reeds geïnstalleerde versie, is het mogelijk dat je de module-map overschrijft. Let op dat dit niet gebeurt!

Nu moet je nog de eigenlijke kernel compileren met "make bzImage".
Als dit commando klaar is zal je de nieuwe kernel vinden in
`/usr/src/linux-2.4/arch/i386/boot/bzImage`

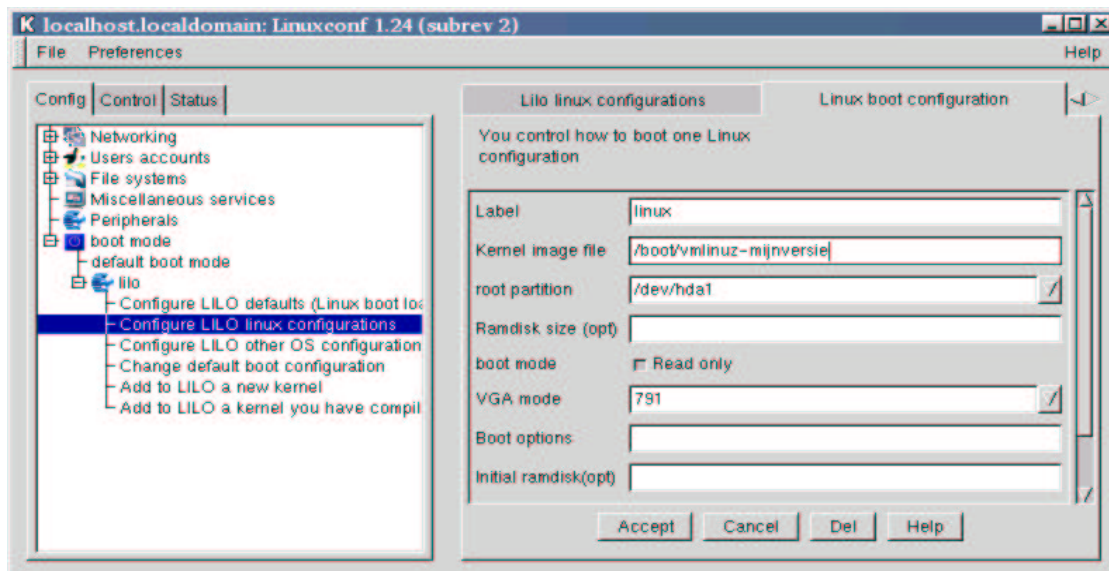
Je kan deze installeren aan de hand van het installatiescriptje dat je in de "`/usr/src/linux-2.4/arch/i386/boot/`" map kan vinden, maar let wel op dat je de oude kernel niet overschrijft (die heb je nodig om te booten als jouw kernel niet zou werken).

Je kan ook het "bzImage" bestand kopiëren naar "/boot" (geef het eventueel een andere naam zoals "`vmlinuz-mijnversie`").

Als laatste stap moet je de "bootloader" configureren zodat deze jouw nieuwe kernel kan opstarten (We gaan ervan uit dat we **LILLO** gebruiken).

Editeer "`/etc/lilo.conf`" met een entry van jouw kernel. Als dit gedaan is bewaar je het bestand en voer je "`/sbin/lilo`" uit.

Deze wijziging kan je ook maken met Linuxconf.



Als je terug van opnieuw wil beginnen, gebruik het commando "make mrproper".

Credits

- <http://www.linuxbelgium.net>
Informatie
- <http://www.linuxdoc.org>
Informatie
- Illiad, <http://www.userfriendly.org>
Cartoons
- <http://lwn.net/Gallery>
Linux logo's
Original penguin logo by Larry Ewing
- Cursus Besturingssystemen/Unix,
Pierre Schavey, Hogeschool Antwerpen
Unix oefeningen
- Cursus CVO Westhoek-Westkust
Lieven Baes
Bestandssystemen
- Taal- en spellingscontrole
Ellen Vleminckx