

Debianguiden - den danske Debian GNU/Linux guide

Version 1.6.0

Debianguiden - den danske Debian GNU/Linux guide: Version 1.6.0

Debianguiden er en installations- og opsætningsvejledning til Debian GNU/Linux, skrevet på dansk. Debian GNU/Linux er et frit og gratis styresystem som bygger på Linux-kernen. Denne guide er skrevet til version 3.0 af distributionen med kodenavnet ``woody".

Indholdsfortegnelse

1. Forord	1
1.1. Ophavsret	1
1.2. Forfattere	1
1.3. Andre bidragydere	1
1.4. Hvordan kan du bidrage?	2
1.4.1. Kan jeg virkelig hjælpe?	2
1.4.2. Hvad kan jeg hjælpe med?	2
1.4.3. Hvordan foregår det i praksis?	2
1.4.4. Hvordan henter jeg kildefilerne?	3
1.4.5. Hvordan arbejder jeg med disse kildefiler?	3
1.4.6. Jeg har lavet en rettelse/ny afsnit - hvad nu?	3
1.4.7. Hvordan genererer jeg guiden selv?	4
2. Introduktion	6
2.1. Hvem er guiden skrevet til?	6
2.2. Debians pakkesystem	6
2.3. Hvorfor Debian?	7
2.4. Hvor kan du ellers finde hjælp?	7
2.4.1. Dokumentation	7
2.4.2. Postlister og forums	8
3. Installation	9
3.1. Installations-kilder	9
3.1.1. Installation over internet	9
3.1.2. Brænding af egen CD	9
3.2. Selve installationen	10
3.2.1. Partitionering af din maskine	10
3.2.2. Valg af drivere/moduler under installationen	11
3.2.3. Netværks opsætning under installationen	12
3.2.4. LILO-opsætning under installationen	12
3.2.5. Spørgsmål efter genstart	13
3.3. Efterfølgende installation af pakker	13
4. Grundsystemet	15
4.1. Netværksopsætning	15
4.2. Brug af apt og dpkg	15
4.2.1. Opsætning af apt	17
4.2.2. Opgradering af distribution via apt	17
4.2.3. Fejlretning og rekonfigurering med dpkg og apt	18
4.2.4. At kun opdatere enkelte pakker	19
4.2.5. Kunsten at holde et woody system opdateret	20
4.3. Rekompilering af programmer	20
4.4. Problemer med danske specialtegn	23
5. Debian som arbejdsmaskine (desktop)	24
5.1. USB-mus	24
5.2. Lydkort	24
5.3. Grafisk brugergrænseflade (XFree86)	25
5.3.1. Skrifttyper i XFree86	25
5.3.2. Window managers	26
5.3.3. Installation af KDE	27
5.3.4. Installation af GNOME	27
5.3.5. GNOME/GTK+-temaer	27
5.4. Drivere til NVIDIA-grafikkort	28
5.5. Internet browser	29
5.5.1. Java og Flash i Mozilla, Galeon og Konqueror	29
5.6. Kørsel af Windows programmer/spil	30
5.6.1. Half-Life	30
6. Debian som server	32
6.1. Webserver	32
6.1.1. Installation og opsætning af Apache	32

6.1.2. Virtual Hosts med Apache (vhosts)	32
6.1.3. PHP	33
6.1.4. Statistik over dine domæner	34
6.2. Databaseserver	34
6.2.1. MySQL	35
6.3. FTP-server	37
6.3.1. Forhindring af at brugere kan se hele fil-hierakiet	37
6.3.2. Forhindring af shell-logins til FTP-brugere	38
6.4. DNS-server	38
6.4.1. Records	39
6.4.2. Installation af bind	39
6.4.3. Opsætning af bind	39
6.4.4. Oprettelse af zonefiler	40
6.5. Firewall og deling af internetforbindelse	41
6.5.1. Installation af iptables	41
6.5.2. Opsætning af iptables - begreber	41
6.5.3. Opsætning af iptables - intro	42
6.5.4. Iptables-eksempler	42
6.6. DHCP-server	44
6.7. Mailserver (Courier)	45
6.7.1. Installation	47
6.7.2. Opsætning af Courier	47
6.7.3. Opsætning af MySQL	50
6.7.4. Test af mail-serveren	50
6.7.5. Stjernealias	51
6.7.6. Courier Mailing List Manager (couriermlm)	52
6.7.7. Courier Webmail Server (SqWebMail)	54
6.8. Proxyserver	55
6.8.1. Installation og opsætning af Squid-proxyserver	55
7. Generel Linux	57
7.1. Basale kommandoer	57
7.2. Pakkede filer (tar, zip)	58
7.3. Manual sider via man	58
7.4. Editorer	59
7.4.1. vi / vim	59
7.4.2. Emacs	60
7.4.3. nano (pico)	61
7.5. Opstartsfiler	61
7.6. Fil-rettigheder og Linux' brugersystem	62
7.6.1. Brugeren, gruppen og de andre	62
7.6.2. Ændring af rettigheder på filer og mapper	63
7.7. Enhedshåndtering	64
7.7.1. Basal brug af mount	64
7.7.2. mount og fstab	64
7.7.3. De mest almindelige linier i fstab	65
7.7.4. Montering af Windows drev	66
7.7.5. Montering af disk-aftryk	67
7.8. At finde rundt på et Linux system	68
7.8.1. Søgning efter filer på systemet	68
7.9. Udskiftning af kernen	69
7.9.1. Kompilering og installation af kerne	69
7.9.2. Installation af prekompileret kerne	72
7.10. Tidsindstillet kørsel af programmer	73
7.11. Fjernkontrol af Linux	74
7.11.1. Brug af SSH	74
7.11.2. PuTTY - en SSH/telnet klient til Windows	76
7.11.3. Kopiering af filer med scp	76
7.11.4. Brug af sftp	76
8. Ofte stillede spørgsmål om... ..	78
8.1. Apt	78
8.2. Netværk	79
8.3. Kerne-kompilering og drivere	80
8.4. X og window managers	81

8.5. Teknik 83

Figurliste

3.1. Debian installation - eksempel på partitionering	10
3.2. Debian installation - installation af moduler	12
3.3. Debian installation - LILO-installation	13
3.4. Debian installation - installation af pakker	14
7.1. En 2.4.23 kernes menuconfig	71

Tabelliste

5.1. Parametre til nvidia-installer kommandoer	28
6.1. Iptables basis elementer	42
7.1. Tabel over pakkesystemer	58
7.2. De ni kategorier af manpages (taget fra "man man")	58
7.3. Strukturen af en man-page	59
7.4. Emacs-kommandoer	61
7.5. Tabel over simple chmod-kommandoer	64
7.6. Oversigt over Linux' fil-hieraki	68

Kapitel 1. Forord

Dansk dokumentation og hjælp til Debian er ikke nemt at finde. [sslug.dk](http://www.sslug.dk/linuxbog/) [<http://www.sslug.dk/linuxbog/>]'s Linux-bog - 'Friheden til at vælge' er ekstrem god, men desværre er den meget alsidig, fordi den retter sig mod alle - specielt RPM-baserede distributioner. Som Debian-bruger vil man helst undgå de generelle ulemper, der er i RPM-baserede distributioner (læs mere om dette i Afsnit 2.3, "Hvorfor Debian?" [7]). Derfor vil denne bog kun koncentrere sig om Debian, dpkg og alle de andre Debian-specifikke ting - der er ingen undtagelser her og der.

Bogen er oprindeligt skrevet af en bunke mennesker på IRC-kanalen #linux.dk, på QuakeNET, hvor Debian var/er meget populært. I dag har mange andre bidraget til guiden. Se Afsnit 1.2, "Forfattere" [1] for de oprindelige forfattere. Afsnit 1.3, "Andre bidragsydere" [1] har en liste med andre bidragsydere igennem tiden.

1.1. Ophavsret

Bogen er udgivet under "Åben dokumentlicens (ÅDL) - version 1.0", som kan læses på <http://www.sslug.dk/linuxbog/licens.html>. Du har bl.a. herved frit lov til at kopiere dette værk uændret på ethvert medium.

1.2. Forfattere

Debianguidens nuværende forfattere:

- Morten Brix Pedersen
- Jacob Emcken

Tidligere forfattere på Debianguiden:

- Kim Ingemann
- Asbjørn Zweidorff Kjær
- Christian Jørgensen

1.3. Andre bidragsydere

Følgende har i mere eller mindre omfang bidraget til udviklingen af Debianguiden:

- Bo Simonsen
- Mads Heckmann
- Anders Rune Jensen
- Allan Jacobsen
- Morten Aaboe
- Morten Aarøe
- Allan Petersen
- Nikolai Beier
- Peter Makhholm

- Joakim Recht
- Rune B. Broberg
- Kasper Rune Søggaard
- Kenneth Dalbjerg
- Marc Cromme
- Michael Kristensen
- Frederik Dannemare
- Tomas Norre Mikkelsen
- Esben Jon Jensen

1.4. Hvordan kan du bidrage?

Denne guide er skrevet af frivillige der vil hjælpe andre ligesindede. Hvis du vil hjælpe os, er du mere end velkommen.

1.4.1. Kan jeg virkelig hjælpe?

Selvfølgelig kan du det men først og fremmest skal du have lyst. For at hjælpe lysten lidt på vej, skal det nævnes, at du kan helt selv kan bestemme, på hvilket teknisk niveau du vil hjælpe og hvor meget tid du kunne tænke dig at bruge på det.

1.4.2. Hvad kan jeg hjælpe med?

For at starte i det helt små, kan man hjælpe ved at bruge Debianguiden og på den måde kvalitetsteste indholdet. Det er i alles interesse at indholdet ikke bare er beskrevet korrekt men også let forståeligt.

Udover fejl i selve indholdet af guiden, er det ikke utænkeligt at der er opstået slå-, stave- og sproglige fejl i skriveprocessen, og dem vil vi selvfølgelig gerne have væk.

Det kunne også være at du bare havde en ide til et ekstra afsnit, eller måske havde fået så meget blod på tanden at du selv havde lyst til at skrive et.

Dokumentation kan være sjovt at skrive, specielt når man skal forklare tingene logisk og uddybende for læseren, dette er en udfordring.

1.4.3. Hvordan foregår det i praksis?

Til de helt små idéer og rettelser til Debianguiden, er det nemmest bare at skrive et indlæg på Debianforum - Fejl og mangler i Debianguiden [<http://www.debianforum.dk/viewforum.php?f=24>].

Hvis du har tænkt dig at afsætte lidt mere tid er det en rigtig god ide at kontakte Debianguiden via postlisten så vi kan koordinere arbejdet inden du går igang. Du kan melde dig på postlisten via denne side [<http://lists.sourceforge.net/lists/listinfo/dkdg-general>].

Efter du har meldt dig på postlisten, kan du sende post til listen via email adressen dkdg-general@lists.sourceforge.net.

Når man skal arbejde på større opgaver med guiden er det nødvendigt at have kildefilerne til guiden. Det er en stor fordel også at have nogle af forskellige værktøjer som bruges til at eksportere guiden til HTML og PDF. I de følgende afsnit vil det blive beskrevet hvordan man gør de forskellige ting.

1.4.4. Hvordan henter jeg kildefilerne?

For at kunne hente kilden til Debianguiden skal man have installeret CVS. CVS er et værktøj der kan håndtere versionsstyring af filer og bruges i mange udviklingssammenhænge. Du installerer CVS på følgende måde:

```
debian:~# apt-get install cvs
```

Nu er CVS installeret.

Stil dig i filsystemet der hvor du ønsker at kildefilerne til guiden skal gemmes (i dette eksempel bruges en brugers hjemmemappe). Der er ikke noget kodeord så når den beder om et, skal du bare trykke enter.

```
bruger@debian:~$ cvs -d:pserver:anonymous@cvs.dkdg.sourceforge.net:/cvsroot/dkdg login
Logging in to :pserver:anonymous@cvs.dkdg.sourceforge.net:2401/cvsroot/dkdg
CVS password:
bruger@debian:~$ cvs -z3 -d:pserver:anonymous@cvs.dkdg.sourceforge.net:/cvsroot/dkdg co
cvs server: Updating guide
U guide/ChangeLog
U guide/Makefile
...
```

1.4.5. Hvordan arbejder jeg med disse kildefiler?

Guiden er skrevet i et XML-format kaldet DocBook. Det er meget let at forstå og logisk at arbejde med. En god måde at starte med at lære Docbook på er at åbne en eksisterende fil, så man kan se hvordan det er gjort.

Her er et par nyttige links i forbindelse med DocBook:

- Linux - Friheden til at skrive dokumentation [<http://www.linuxbog.dk/dokumentation/bog/index.html.php>]
- DocBook XSL: The Complete Guide [<http://www.sagehill.net/docbookxsl/index.html>]
- DocBook reference [<http://www.faqs.org/docs/docbook/html/>]
- DocBook eksempler (ligger i CVS) [<http://dev.w3.org/cvsweb/XML/test/Docbook/#dirlist>]
- DocBook parametre [<http://docbook.sourceforge.net/release/xsl/current/doc/html/index.html>]

Hvis du har brug for yderligere hjælp kan du skrive til postlisten.

1.4.6. Jeg har lavet en rettelse/ny afsnit - hvad nu?

Hvis du har lavet nogle rettelser til en fil, sender du en patch til postlisten. En patch er en fil, der viser forskellen imellem 2 filer - den originale og din egen redigerede. For at lave en patch, bruges denne kommando:

```
bruger@debian:~$ diff -u [original_fil] [din_redigerede_fil] > [patch_til_guide_fil]
```

Det kunne eksempelvis se sådan her ud:

```
bruger@debian:~$ diff -u forord.xml ny_forord.xml > forord.xml.patch
```

Herved vil du have en patch med filnavn `forord.xml.patch`

Når din patch er færdig kan du e-maile den til postlisten. Men hvis den er stor (over 40kb) bør du smide den på nettet så den kan downloades, da der er en grænse på størrelsen af vedhæftninger på mailing listen.

1.4.7. Hvordan genererer jeg guiden selv?

Mens du sidder og arbejder vil du sikkert gerne se resultatet af din indsats. For at du kan se hvordan dine ændringer tager sig ud skal du eksportere DocBook til f.eks. HTML eller PDF. Eksport til begge formater vil blive beskrevet i de følgende afsnit.

Uanset om du vil eksportere til HTML eller PDF skal du have installeret nogle grundpakker der kan håndtere DocBook.

```
debian:~# apt-get install make xsltproc
debian:~# apt-get install docbook-xml docbook-xsl sgml-base sgml-data
```

1.4.7.1. Eksport til HTML

For at eksportere til HTML behøver man kun at have grundpakkerne installeret, som beskrevet ovenfor. Du kan eksportere Debianguiden til HTML ved at benytte det medfølgende 'make'-script.

```
bruger@debian:~/guide$ make html
```

Nu vil du kunne finde guiden i HTML-format i ~/guide/html/.

1.4.7.2. Eksport til PDF

Lige netop PDF kan volde lidt problemer, for det er nødvendigt at benytte et værktøj der hedder FOP, og FOP findes ikke i stable.

Hvis du benytter testing/unstable, kan du installere FOP via apt-get:

```
debian:~# apt-get install fop
```

Dog vil billeder i de generede PDF'er ikke virke, før du har installeret Jimi-biblioket, som desværre ikke findes som Debianpakke. Istedet henter du Jimi (jimi1_0.zip) fra Suns hjemmeside [<http://java.sun.com/products/jimi/>].

```
debian:~# unzip jimi1_0.zip
debian:~# cp Jimi/JimiProClasses.zip /usr/share/java/jimi-1.0.jar
```

Hvis du benytter woody/stable, skal man hente FOP fra The Apache XML Project: Fop [<http://xml.apache.org/fop/>]. Bare rolig det er nemt at afinstallere igen, og dette vil selvfølgelig også blive beskrevet her.

I det følgende eksempel hedder filen fop-0.20.5-bin.tar.gz og er gemt i roots hjemmemappe.

```
debian:~# cd /usr/lib
debian:/usr/lib# tar -zxvf ~/fop-0.20.5-bin.tar.gz
debian:/usr/lib# ln -s /usr/lib/fop-0.20.5/fop.sh /usr/bin/fop
```

Derefter skal du hente Jimi (jimi1_0.zip) fra Suns hjemmeside [<http://java.sun.com/products/jimi/>].

```
debian:~# unzip jimi1_0.zip
debian:~# cp Jimi/JimiProClasses.zip /usr/lib/fop-0.20.5/lib/JimiProClasses.jar
```

Hvis man vil afinstallere FOP (og Jimi) igen gøres det ved at slette mappen og symlinket man har oprettet.

```
debian:~# rm -r /usr/lib/fop-0.20.5
debian:~# rm /usr/bin/fop
```

Kapitel 2. Introduktion

Debian er ikke som de andre distributioner. Hvorfor?

Selvom det måske ikke umiddelbart har din interesse, så er Debian ikke en kommerciel distribution. Den bliver med andre ord ikke lavet af et firma, som har stramme deadlines. Den bliver i stedet udviklet af frivillige mennesker fra hele verden. Dette er folk, som virkelig går op i deres "arbejde", og de vil altid levere det bedste. Linux-folk generelt har en speciel mentalitet, der går på, at alting skal være "korrekt" og "fejlfrit" - sådan er det også med Debian-udviklerne.

Der er over 900 Debian-udviklere verden over. En meget stor del af arbejdet som Debian-udvikler består for de flestes vedkommende i at vedligeholde software, som bliver 'pakket' i mindre dele, som gør det let og overskueligt at fjerne og tilføje dele af et system. En 'pakkeansvarlig' Debian-udvikler har ansvaret for, at hans pakker bliver holdt opdateret i takt med at nye versioner af den pågældende software frigives. Derudover er der generelt vedligeholdelsesarbejde af pakkerne, såsom forbedringer til installationen og standardopsætningen af en given pakkes indstillinger. Via Debians fejlrapporteringssystem [<http://www.dk.debian.org/Bugs/>] kan du selv se de fejl, som berører pakkerne lige nu. Du kan selv rapportere nye fejl, hvis du har fundet nogen, eller måske endda rette eksisterende fejl. På den måde har alle muligheden for at være med til at gøre Debian til en bedre distribution.

Debian er ikke umiddelbart en nem distribution at starte ud med. Ting som automatisk hardware-detektion, som du finder i andre distributioner, findes ikke i Debians nuværende stabile udgivelse (dog med få undtagelser).

Ligeledes er der heller ikke det store udvalg af grafiske opsætningsprogrammer, men fortvivl ikke - alt dette er ikke nødvendigvis en dårlig ting, idet det nemlig indbyder til, at man som bruger selv lærer lidt mere om, hvordan tingene fungerer "under hjelmen" på ens Linux-system.

2.1. Hvem er guiden skrevet til?

Debian kan køre på mere end bare en PC. Debian 3.0 ("Woody") understøtter i alt 11 processor-arkitekturer. Disse er: "Alpha", "ARM", "HP PA-RISC", "Intel x86", "Intel IA-64", "Motorola 680x0", "MIPS", "MIPS (DEC)", "PowerPC", "IBM S/390", "SPARC". En processor-arkitektur har noget at gøre med, hvordan maskinens CPU er opbygget. Du kender måske kun til én processor-arkitektur - nemlig "Intel x86" - som er den, vi hyppigt kalder for PC'en. De mest kendte x86 CPU'er er de, som AMD og Intel producerer. Denne guide er skrevet med hovedvægt på PC-brugeren, da det er svært at tage hensyn til alle 11 processor-arkitekturer.

Udover at være ejer af en PC, er guiden skrevet til dig, som falder ind under en af disse kategorier:

- Du vil i gang med Linux for første gang.
- Du har prøvet andre Linux- eller UNIX-styresystemer og vil nu prøve Debian.
- Du er Debian-bruger og bruger guiden som opslagsværk til eventuelle problemer.

Det er også vigtigt, at du er den type, som tager tiden til at læse om tingene, før du hopper ud i den store Debian/Linux verden. Det er svært (hvis ikke umuligt) at lære Linux ved at prøve sig frem uden dokumentation. Det er vigtigt at *læse* dokumentationen og ikke bare skimme let henover den.

2.2. Debians pakkesystem

En af Debians stærkeste sider, og nok også den mest omtalte, er pakkesystemet. Hvis du er tidligere bruger af en RPM-baseret distribution (som f.eks. Red Hat eller Mandrake), så er du højst sandsynligt stødt ind i de utrolig mange afhængighedsproblemer, der kan opstå, når der skal installeres nye pakker. Nyere versioner af RPM-baserede distributioner er omsider ved at råde bod på dette problem, og problemet er ikke så udtalt længere, som det var engang.

Lad os prøve at lave en direkte sammenligning. Lad os sige, at vi skal installere pakke FOO, som ikke er med i standardinstallation. Vi starter med Red Hat.

Først går vi ind på ftp.redhat.com (eller på vores CDROM), og henter `FOO.rpm`. Herefter går vi ud af FTP'en,

su'er til root, og skriver **rpm -i FOO.rpm**. RPM fortæller os nu at FOO kræver BAR pakken. Ok, vi går ind på ftp'en igen, henter BAR.rpm, og laver en **rpm -i BAR.rpm**. RPM fortæller os nu at BAR kræver libXYZ.so filen, men fortæller os ikke hvilken pakke den ligger i. Dette betyder at vi må gå ud på rpmfind.net [<http://www.rpmfind.net>] og hente en tredje-parts RPM hvor libXYZ.so ligger i.

Og sådan kan man blive ved. Der er mange større problemer i RPM, men det her er den mest generelle som folk irriterer sig over. Lad os nu prøve at gøre det samme med en Debian maskine:

Vi skal installere FOO, så derfor skriver vi **apt-get install foo** - apt tjekker listen over pakker, laver et afhængigheds (dependency) træ og fortæller os at den også skal hente BAR og superXYZ pakkerne, for at vi kan installere FOO. Dette siger vi ja til, og herefter henter apt automatisk pakkerne ned fra nettet/cdrom, installerer dem - og spørger dig i nogen tilfælde om et par spørgsmål vedr. opsætning af pakkerne. (Det er endnu et problem ved RPM, der er ikke mulighed for at spørge brugeren om noget ved installatin af pakken, så man er efterladt med standardindstillinger, som ikke altid virker). Nå, men det var det - vi har alle vores pakker installeret, og det sparede os en masse tid.

Ok, det er ikke en helt fair sammenligning. **apt-get** er faktisk ikke et pakkesystem, men et frontend til det egentlig pakkesystem, **dpkg**. Men vi håber at du forstår pointen.

Det var kort nogen fordele ved Debians pakkesystem, der er mange flere - men for at du ikke skal falde helt i søvn, så stopper vi nu.

2.3. Hvorfor Debian?

Dette er delvist besvaret i de foregående afsnit, men lad os summe lidt op:

- Godt pakkesystem. For brugeren er det utroligt nemt at installere software-pakker.
- Kvalitet. Udviklet af mennesker der går op i det - og tingene bliver ikke forhastet fordi en stram deadline skal overholdes.
- Debian er 100% 'free', (her hentyder vi til frihed og ikke pris). Debian har en social kontrakt [http://www.dk.debian.org/social_contract] som definerer den form for frihed som distributionen skal have.
- Debian gør tingene mere "rigtigt", et holdnings-spørgsmål måske. Men Debian prøver ikke at tilfredsstille den ny-ankomne Windows bruger, men i stedet prøver at vise hvordan tingene skal gøres 'The Unix Way'.
- Nyeste software, nemt. Ved at opgradere til unstable eller testing kan du få de nyeste pakker ind ad døren hver aften.
- Stor bruger-base. Debian bruges af rigtig mange over hele verden. Over IRC eller HOWTO's på nettet (som denne) er der stor hjælp at hente hvis du kører Debian.
- Stor sikkerhed. Debian bestræber sig på at holde distributionen så sikker som muligt [<http://www.dk.debian.org/security/>]. Normalt varer det kun op til 48 timer før en ny pakke som ordner et sikkerhedshul er ude.

2.4. Hvor kan du ellers finde hjælp?

Debianguiden er langt fra den eneste mulighed for at finde dokumentation eller hjælp til opsætning af et Debiansystem.

2.4.1. Dokumentation

Nyttige kilder til dokumentation.

- Den officielle installationsvejledning til Debian GNU/Linux. [<http://www.dk.debian.org/releases/stable/installmanual>]

- Debianplanet.org - et site hvor nybegyndere kan stille spørgsmål, der foregår også mere tekniske diskussioner [<http://www.debianplanet.org>]
- Debianhelp.org - et site som ovenstående, dog mere aktivt med flere spørgsmål om relevante emner. [<http://www.debianhelp.org>]

2.4.2. Postlister og forums

Udover den tørre dokumentation, kan du også få hjælp i mere levende forums. Der findes danske postlister vedr. Debian, samt online forums på nettet.

- Debianforum - et webbaseret hjælpeforum. Nemt at gå til. [<http://debianforum.dk/>]
- Den officielle danske Debianbruger postliste (debian-user-danish). [<http://lists.debian.org/debian-user-danish/>]
- Den officielle postliste for danske oversættelser i Debian. Hvis du ser fejl i danske oversættelser, skal du kontakte denne postliste. [<http://lists.debian.org/debian-l10n-danish/>]
- Nyhedsgruppen dk.edb.system.unix kan også bruges til at stille spørgsmål. Brug din yndlings nyhedsgruppeklent.
- Henrik Bøgghs liste over danske UNIX-relaterede IRC-kanaler, hvor du kan få hjælp. [http://lagengymnastik.dk/~henrik/geek/irc_kanaler.html]

Kapitel 3. Installation

Installationen af Debian kan være svær hvis man ikke har voldsomt meget erfaring med forskellige computer systemer. Her vil vi prøve at beskrive en installationsproces step-by-step. Hvis du vil gå mere i dybden med installationen, eller har nogen ting du undrer dig over, så henviser vi til den officielle installations-vejledning [<http://www.dk.debian.org/releases/woody/installmanual>].

3.1. Installations-kilder

Der er en del måder hvorpå du kan installere Debian, find selv hvilken metode der passer dig bedst.

Installation over internettet, er foretrukken hvis du har en hurtig/fast internet forbindelse, se Afsnit 3.1.1, "Installation over internet" [9].

Hvis du ikke har en hurtig linje, men stadig har adgang til en fast linje, så kan du hente ISO'en (et CD image), og brænde det ned på en CD, se Afsnit 3.1.2, "Brænding af egen CD" [9].

Sidste løsning, er at købe Debian, det er ekstremt billigt - vi kan anbefale [linuxpusher.dk](http://www.linuxpusher.dk) [<http://www.linuxpusher.dk>].

3.1.1. Installation over internet

Hvis du skal installere Debian over internet, så kræver det at du har noget at boote installationen på. Dette kan enten være en bootable CD som du selv har lavet, eller boot-disketter.

For at køre en installation via floppy bootdisks, så skal du have fat i følgende images: `rescue.bin`, `root.bin`, `driver-1.bin`, `driver-2.bin`, `driver-3.bin`, `driver-4.bin`. Bootdiskene kan fås fra samtlige Debian mirrors [<http://www.dk.debian.org/distrib/ftplist>], i skrivende stund ligger de her: <http://ftp.se.debian.org/debian/dists/woody/main/disks-i386/>

Bemærk! - du skal hente de images der ligger under `bf2.4` mappen hvis du vil installere på en 2.4 kerne - dette kan anbefales på de fleste systemer, og er krævet på systemer hvor din hardware ikke er understøttet i de normale images (som bruger en 2.2 kerne).

For at skrive filerne ned på diskette, skal du til Windows bruge programmet `rawrite2.exe`. Det ligger i `dosutils/` mappen, på samme FTP.

Start herefter `rawrite2.exe` (vi går her ud fra at du kører Windows), og skriv dine images ned på disketter. Herefter kan du starte installationen ved at boote med `rescue.bin` disken først.

Hvis du sidder ved en Linux maskine allerede, og vil skrive en `.bin` til diskette, så er det simpelt med kommandoen `dd if=rescue.bin of=/dev/fd0 bs=1440k`.

3.1.2. Brænding af egen CD

En anden installations mulighed, er at brænde Debian ned på CD selv. I skrivende stund ligger der ISO images på denne adresse: http://ftp.se.debian.org/debian-iso/images/3.0_r5/.

På sitet vil du finde adskillige CD images. Der er samtidig mange forskellige arkitekturer som der kan hentes ISO'er til. Til almindelige PC'ere, skal man have fat i `i386` mappen. Kun første CD er krævet for en installation. Du skal være opmærksom på at det kan være en fordel at hente NON-US CD'en. Her ligger ting som ikke må eksporteres fra USA. F.eks. `proftpd`. I skrivende stund hedder dette image `debian-30r3-i386-binary-i386-1_NONUS.iso`. På CD2 og CD3 ligger der ekstra pakker som du (i de fleste tilfælde) ikke har brug for. Du kan altid hente dem med **apt-get** senere.

Når du skal brænde ISO filen ned på CDRom skal den ikke bare brændes over som en normal fil. Du skal have fat i et program som understøtter at skrive ISO'er til en CD. Heriblandt kan nævnes `Adaptec Easy CD Creator`, `Nero`, `CDRWIN` til Windows. På et Linux system kan du bruge `cdrecord`. Du kan finde flere informationer om hvordan du brænder et Debian CD-image (eller CD-images generelt) i Debian CD FAQ

[<http://www.dk.debian.org/CD/faq/#record-windows>]'en.

Når du har brændt ISO'en ned på en CD, så er det bare at putte CD'en i drevet og boote maskinen. Hvis maskinen ikke vil boote på den kan det være fordi at du ikke har slået det til i BIOS. Hvis dette også fejler, så bliver du nødt til at bruge en boot-diskette til at boote installationen, se Afsnit 3.1.1, "Installation over internet" [9].

3.2. Selve installationen

Du starter installationen ved enten at boote på din CD eller at boote på din boot-diskette (rescue disken).

Noget af det første som man ligger mærke til når man starter sin Debian installation, er at der her ikke er tale som en fancy grafisk brugerflade til installationen som ved andre distributioner, men blot et simpelt semigrafisk, tekst-baseret menu-system.

Systemet giver meget fleksibilitet for den avancerede bruger og er brugervenlig på mange måder - men den er stadig svær hvis man ikke kender alle begreberne. I de kommende afsnit vil de punkter som anses for svære, gennemgås.

3.2.1. Partitionering af din maskine

For et almindeligt arbejdsstation-system anbefales det at sætte mindst 1 eller 2 GB plads af til Debian. Hvis du sætter en web-server eller lign. op behøver det ikke at være så meget.

Under installationen møder du partitionerings-programmet **cfdisk** - det er her hvor du skal partitionere harddisken op i mindre stykker. Partitionering i Linux er meget anderledes end i Windows. Man tildeler ikke partitioner eller harddisk drev-bogstaver, men derimod fortæller man dem hvor på filesystemet de skal monteres.

Efter valg af **Partition a Hard Disk**, skal du vælge den harddisk, du ønsker at oprette dine Debian partitioner på. Her er det vigtigt at vide hvilken controller din harddisk sidder på hvis man har flere harddiske. Primary IDE1 controller er `/dev/hda`, Secondary IDE1 controller er `/dev/hdb` o.s.v.

Til din første installation af Debian, vil vi råde dig til at lave følgende 3 partitioner:

- `/boot` - en lille 10-15 mb partition hvor din kerne og andre boot-specifikke ting ligger på i starten af drevet.
- `swap` - i Linux opretter man en swap partition i stedet for at have en global swap fil, din swap partition skal være alt fra 64mb til 512mb afhængende af dine ram og harddisk plads.
- `/` - din rodpartition, denne partition skal være den største partition, og egentlig udgøre resten af pladsen på din harddisk.

Figur 3.1. Debian installation - eksempel på partitionering

```

cfdisk 2.11n

Disk Drive: /dev/hda
Size: 2097152000 bytes
Heads: 128 Sectors per Track: 32 Cylinders: 1000

Name      Flags      Part Type  FS Type      [Label]      Size (MB)
-----
hda1     Boot      Primary   Linux ext2    20.98
hda2                    Primary   Linux ext2    1700.80
hda3                    Primary   Linux swap    375.40

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Toggle bootable flag of the current partition_

```

Når **cfdisk** er startet, bruger du piletasterne til at rykke rundt med. For at oprette en ny partition skal du vælge **New**. Hvorefter du bliver spurgt om det skal være en primary eller logical partition - her vælger du primary. Dernæst vælger du hvor mange MB din partition skal være. Til den første partition, din boot-partition, angiver du at den skal være 15 MB og trykker **Enter**. Nu skal du så vælge om du vil have partitionen til at ligge i starten eller i slutningen af harddisken. Du vælger her **Beginning**.

Du vil nu se en liste over partitioner. Her vil din 15 MB's partition fremgå, og under den er en "partition" mere som er din frie harddiskplads. Du skal markere din boot-partition som Bootable, som du gør simpelt nok ved at trykke på knappen **Bootable**.

Du skal nu lave partition nr. 2. Du vælger nu med op- og ned-piletasterne din frie harddisk plads og vælger **New**. Samme procedure som før skal nu gentages.

Denne gang skal swap-partitionen laves, den kan som før nævnt være alt fra 64-512 mb. Når du har oprettet partitionen er det vigtigt at du vælger **Type** og skriver '82'. '82' er typen for et **Linux swap** drev.

Til sidst skal din rodpartition laves, du bruger samme procedure som ved din første 15mb partition, men her tager du bare resten af harddisk pladsen der er tilbage. Husk at check i **FS Type** feltet at denne partition er en normal Linux (ext2) partition.

Du er nu færdig med at partitionere og skal afslutte ved at trykke på **Write** efterfulgt af **Quit**.

Du kommer tilbage til menuen - og nu skal du formatere og aktivere SWAP partitionen. Derfor vælger du **Initialize and Activate a Swap Partition**. De næste par spørgsmål er trivielle og vil ikke gennemgås her.

Du skal nu formatere dine Linux partitioner (ext2 filsystemer). Du vælger derfor **Initialize a Linux Partition**. Dernæst vælger du den partition du vil formatere (du skal formatere alle partitioner) med piletasterne og trykker **Enter** når du er på den partition der skal formateres. Den første partition du vil formatere er partition nummer 3, da denne skal være din / (root) system. Til sidst spørger den om du vil montere den som dit root filesystem ("/"). Her vælger du **Yes**.

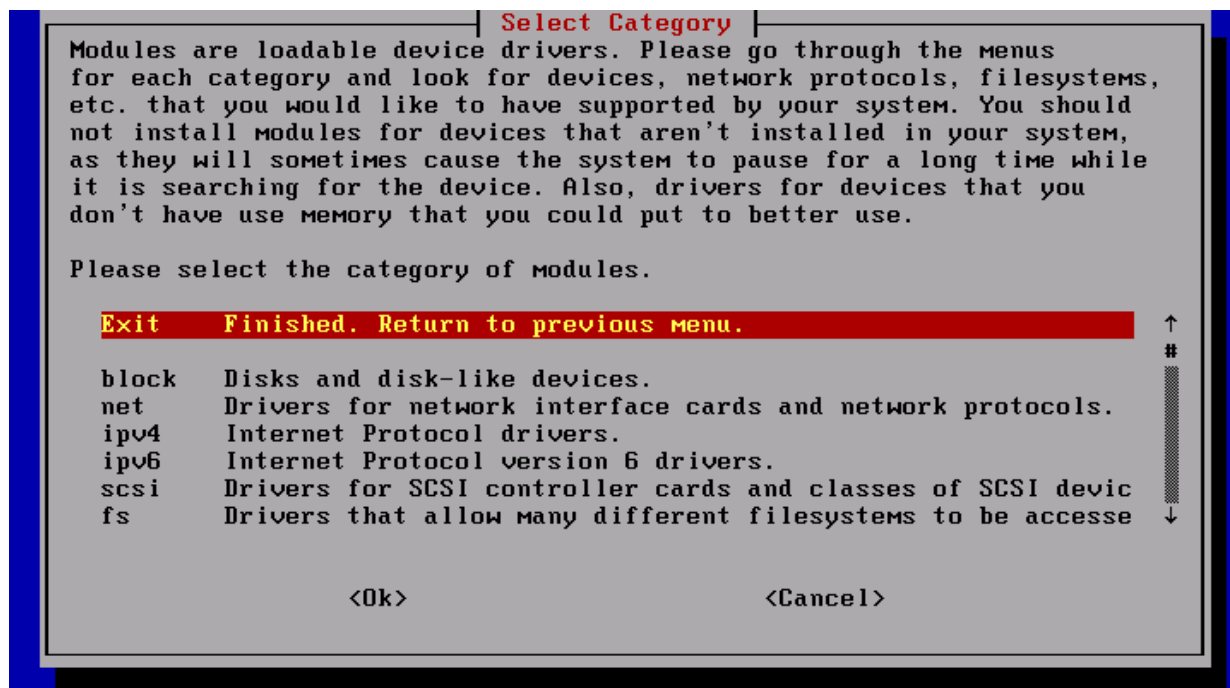
Nu skal du montere den sidste partition, som jo var din /boot partition på 15 mb. Det gør du i menuen ved at trykke **Initialize a Linux Partition**. Gentag nu processen som tidligere, indtil du kommer til en ny skærm hvor du skal vælge hvad dit mount point skal være. Den første partition (som regel: /dev/hda1) skal du montere som /boot.

3.2.2. Valg af drivere/moduler under installationen

Næste "kritiske" punkt i installationen er når du kommer til punktet **Configure Device Driver Modules**.

Det er her du installerer drivere til din hardware. I modsætning til mange andre distributioner, finder Debian ikke selv ud af hvilken hardware der er i maskinen. Det er derfor vigtigt at man er bekendt med sit system, og ved præcis hvilket netkort, grafikkort o.s.v man har. Det første der sker er, at systemet spørger om man vil installere drivere fra en diskette eller lign. Dette vil dog ikke være nødvendigt på nuværende tidspunkt, da du uden problemer kan gøre dette senere. Desuden er det meste hardware i dag understøttet som standard i kernen. Derfor skipper du punktet om at hente drivere fra diskette.

Figur 3.2. Debian installation - installation af moduler



Du vil nu få en liste over hvilke typer drivere, du kan installere. Normalt vil man ikke have behov for at ændre i **block**, **cdrom**, **fs** og **ipv6** menuerne, men man kan dog godt hvis man ønsker det. Under **net** kan du vælge hvilke netkort du ønsker understøttelse af i kernen. For at installere et modul, skal man markere det med piletasterne og trykke på **Enter**. Man vil så blive spurgt om man ønsker at installere modulet som en del af kernen. Når man har sagt ja til at installere en modul som en del af kernen, vil man kunne angive ekstra **Command-Line arguments**. Det er dog meget sjældent at man ikke kan klare sig uden disse, så derfor trykker du bare OK her. Nu vil der så, hvis alt er gået vel, komme en besked på en sort skærm om at modulet er blevet installeret. Når alle de ønskede moduler er loadet, vælger du **exit** så du kommer ud i hovedmenuen og endnu en gang **exit** for at afslutte modul-konfigurationen.

3.2.3. Netværks opsætning under installationen

Hvis du fik loadet et modul til dit netkort, vil du blive spurgt om at sætte netværket op. Punktet hedder **Configure the Network** i installationen. Netværket er relativt simpelt at sætte op, hvis du kender noget til netværk generelt.

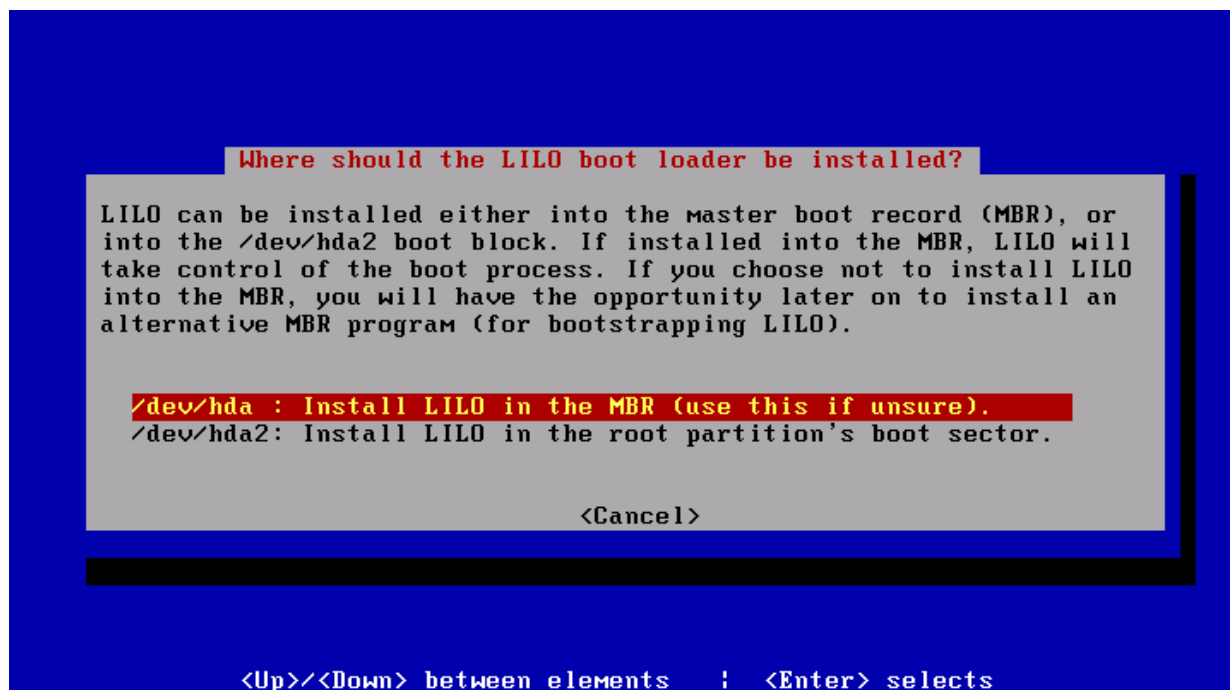
Du bliver først spurgt om du vil bruge BOOTP/DHCP (automatisk ip hentet fra router eller anden server). Hvis du siger nej til dette, skal du selv indtaste din ip. Hvis du ikke kender noget til hvilke ip'er du bør bruge, kan du nøjes med standard indstillingerne.

3.2.4. LILO-opsætning under installationen

Efter base-systemet er hentet fra CD, internet eller diskette - kommer du til menupunktet **Make System Bootable**. Her bliver du spurgt om hvor bootloaderen (LILO) skal installeres på harddisken. På næste skærm

(som ses forneden) skal du vælge hvor LILO skal installeres. Her er det normalt bedst at vælge **Install LILO in the MBR**.

Figur 3.3. Debian installation - LILO-installation



Som sidste punkt i installationen kan du vælge **Make a Boot Floppy**. Det er anbefales-værdigt at altid have en boot-diskette klar, specielt hvis du er en ny Linux bruger. Men det er ikke påkrævet for installationens videre gang.

Efter installationen af LILO og boot-disken, kan du genstarte systemet ved hjælp af punktet **Reboot the System**.

3.2.5. Spørgsmål efter genstart

Efter systemet er genstartet, skal du til at besvare en masse spørgsmål. De trivielle spørgsmål vil ikke blive gennemgået her.

Du bliver spurgt om du ønsker at aktivere MD5 kryptering og shadow-fil til dine brugeres kodeord. Dette giver kun en bedre sikkerhed - så her kan du trygt svare ja.

Herefter skal du angive hvilket kodeord du ønsker til din root-bruger. root-brugeren er systemets administrator og det er vigtigt at du ikke bruger samme kodeord her som dine almindelige brugere på systemet.

Du vil bagefter blive spurgt om du vil tilføje ekstra brugere til systemet, det skal du altid sige ja til, eftersom normalt arbejde på din Debian-box skal foregår som en almindelig bruger og IKKE root.

Du vil også blive spurgt om systemet skal fjerne PCMCIA pakken, dette er ønsket hvis du ikke har en bærbar maskine. Du får samtidig muligheden for at installere systemet via PPP. PPP er en almindelig modem forbindelse (d.v.s. ikke ADSL eller en anden fast-opskobling).

Til sidst skal du i gang med pakke installation, dette er dækket i Afsnit 3.3, "Efterfølgende installation af pakker" [?].

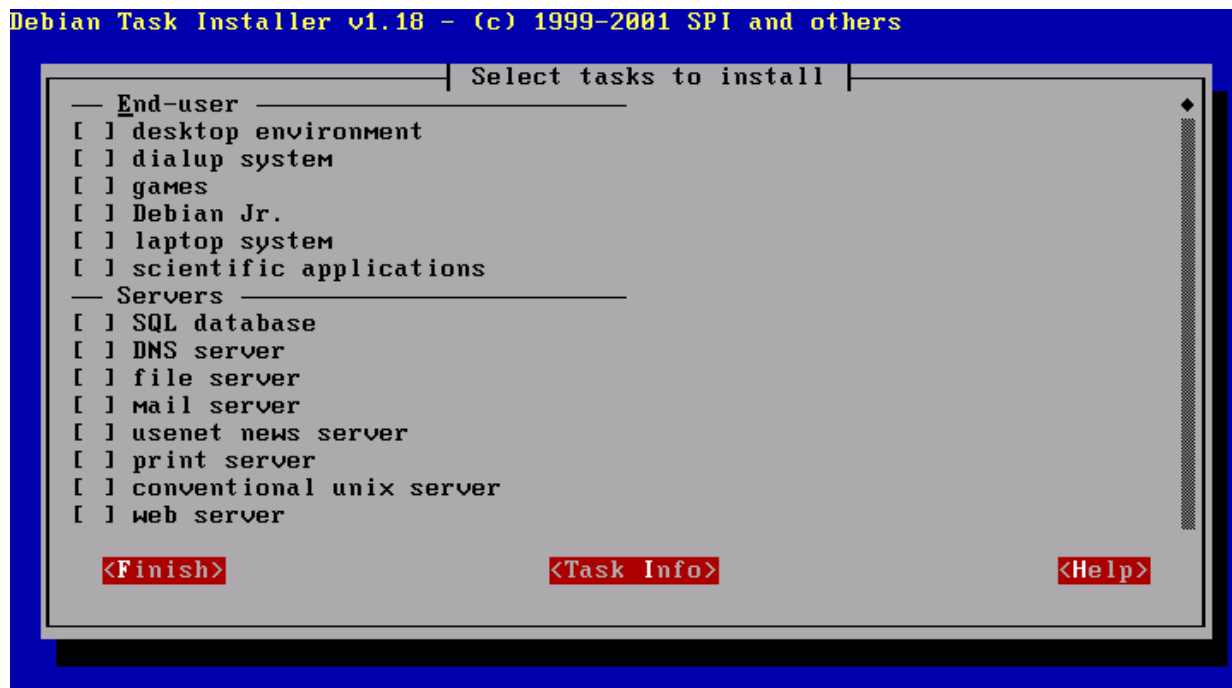
3.3. Efterfølgende installation af pakker

Som næste led i installationen vil du blive spurgt hvilken kilde du vil installere pakker fra. Her vælges CD-ROM

hvis du installerede fra CD-Rom eller internet hvis du installerede via internet o.s.v.

Efter du har angivet hvilke pakker du vil have og valgt mirror, vil du blive spurgt om du skal starte **tasksel**. Tasksel er den simple udgave af pakke installationen og kan kraftigt anbefales. I tasksel vælger du de pakker "kategorier" du vil have. F.eks. vælger du "X window system" hvis du vil have en grafisk brugergrænseflade.

Figur 3.4. Debian installation - installation af pakker



Efter du har afsluttet tasksel, vil blive spurgt om du vil starte **dselect** - dette er et sværere program at vælge pakker med og skaber normalt mere forvirring end gavn. Det er normalt nemmere at holde sig til at bruge apt til at installere pakker.

Under installationen af pakker vil man opleve endnu en af de stærke sider i Debians pakkesystem. Fremfor at installere alle programmer med standard indstillinger, vil man i stedet under installationen blive spurgt om vigtige spørgsmål.

Du vil blive spurgt en masse spørgsmål, om mange forskellige ting, prøv at gør dit bedste ved at læse beskrivelserne grundigt og først efter det svare på dem. Det kan også være klogt at have en Linux mand ved siden af sig, til hjælp.

Efter du har besvaret alle spørgsmål og alle pakker har installeret færdigt er din installation af Debian klar! Held og lykke.

Kapitel 4. Grundsystemet

Dette kapitel indeholder information om opsætning af Debians grundsystem.

4.1. Netværksopsætning

For mange af de lidt ældre Linux brugere, som f. eks. har brugt Red Hat før, vil de bemærke at i Debian, ikke findes programmet **'netconf'** til at sætte netværks-konfigurationen op. Den vane vil de helt nye Debian brugere slippe for.

Det første man skal gøre er at load modulet(driveren) til sit netkort. Dette gøres via Debians **modconf**. I **modconf** skal du gå ind i 'net' sektionen og vælge modulet til dit netkort. Hvis du ikke ved hvilket netkort du har, så er der ikke andet for at finde ud af det - enten i din manual eller kigge på chippen.

Når modulet er indlæst, skal du redigere filen `/etc/network/interfaces` med din yndlings editor. Denne fil indeholder al information som kommandoerne **ifup** og **ifdown** skal bruge. Der står informationer såsom: IP-adresse, gateway, subnet maske, osv.

Filen kan også indeholde kommentarer. Det ses ved at i starten af linjen er der en '#' (havelåge).

I filen findes alle de forskellige 'interfaces' (forkortet '**iface**'), som enkeltvis definerer en netværksforbindelse. Eksempel:

```
auto eth0
iface eth0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
```

Den første linje, **auto eth0** betyder at netkortet vil blive initialiseret automatisk ved boot. Resten definerer vores netkort **eth0**. Vi fortæller at vi selv vil angive en ip ved at skrive **static**. Dette netkort bliver sat op med ip-adressen 192.168.1.1 og netmasken 255.255.255.0. De forskellige elementer som kan bruges, **address**, **netmask**, **broadcast**, **network** og **gateway** står beskrevet i **man interfaces**.

Et andet eksempel ser således ud:

```
auto eth0
iface eth0 inet dhcp
```

dhcp betyder Dynamic Host Configuration Protocol og betyder at netkortet automatisk vil prøve på at få en IP-adresse af en DHCP-server på netværket. Dette er f. eks. tilfældet hvis du forbinder dig via en router til internettet hvor routeren deler IP-adresserne ud.

Efter `/etc/network/interfaces` er opsat færdigt, så kører du **ifup eth0**. Dette vil initialisere netværks indstillingerne for netkortet. Du kan nu kigge med **ifconfig** om netkortet er oppe. Hvis netkortet ikke er, så sørg for at kig i **dmesg** om netkortet blev fundet korrekt.

Normalt skal du også angive hvilket nameservere som du bruger. Disse er angivet af din internet udbyder. Opret filen `/etc/resolv.conf`. Heri skal du definere ip-adresserne på de navneservere som du skal bruge. Her er et eksempel:

```
nameserver 1.2.3.4
nameserver 4.3.2.1
```

Hvis du har problemer med netværks opsætning, se da man-siden for `interfaces`.

4.2. Brug af apt og dpkg

Når du arbejder med Debian, vil du støde på at man skal bruge **dpkg**. dpkg er Debians egentlige pakkesystem. Det

kan ske at du f. eks. henter Opera, som ikke ligger som prekompileret i Debian. Derfor bliver du nødt til at installere den manuelt, dette gøres ved følgende kommando:

```
debian:~# dpkg -i pakkenavn.deb
```

Hvor pakkenavn.deb er navnet på pakken. Du kan fjerne den igen med:

```
debian:~# dpkg -r pakkenavn
```

Men normalt når du installerer pakker, så vil den ligge i det arkiv som Debiansystemet kender, det er nemlig det arkiv som er defineret i `/etc/apt/sources.list`. Her kan man selv tilføje flere kilder. Programmet **apt-get** vil kigge her når den skal installere pakker. f. eks.

```
debian:~# apt-get install mozilla
```

vil installere Mozilla pakken. Du behøver ikke at kende det fulde filnavn, eller hvor den ligger henne. apt-get vil hente den enten fra din CD, fra web eller lignende. Det kommer helt an på indholdet af din `/etc/apt/sources.list`.

Pakken kan også fjernes med **apt-get**:

```
debian:~# apt-get remove mozilla
```

Du kan også bygge kildekode pakker så de er optimeret til dit eget system, f. eks. hvis du har en i686 så vil du undgå at en speciel pakke er kompileret til i386. Det kan du gøre således:

```
debian:~# apt-get --compile source mozilla
```

apt-get vil nu hente sourcen til Mozilla og kompilere den for os. Herefter kan den installeres med **dpkg**. Se i øvrigt afsnittet omkring rekompilering af pakker for mere information.

Hvis du får behov for at se hvilke pakker du faktisk har installeret, kan det gøres på følgende måde:

```
debian:~$ dpkg --get-selections
```

Det smarte ved det, er at du kan ligge det ud i en fil ved **dpkg --get-selections > fil**. Det kan være ret nyttigt at gemme denne fil, da hvis der sker det at din maskines harddisk ryger eller lign. og vil "re-etablere" dit system, kan du gøre det ved **dpkg --set-selections < fil**.

Som en del af apt findes et værktøj der hedder **apt-cache**, som du senere vil skifte bekendtskab med. Det program kan søge i pakke databasen:

```
debian:~$ apt-cache search browser
```

hvor browser her er vores søgestreng. Derudover kan den vise informationer om en pakke, det er ret nyttigt hvis du vil se hvad denne pakke egentlig gør:

```
debian:~$ apt-cache show mc
```

for at se hvad pakken mc gør. Sluttelig kan man inspicere status og installationsplaner for en given pakke med

```
debian:~$ apt-cache policy pakkenavn
```


4.2.1. Opsætning af apt

Når du skal hente pakker med `apt`, så kan det gøres fra en del forskellige kilder, f. eks. FTP, HTTP, NFS, CDROM og floppy.

Der er 2 måder hvorpå du kan sætte `apt` op, det kan enten gøres manuelt via `/etc/apt/sources.list`, eller via programmet **apt-setup**. Førstnævnte er foretrukket for fleksibilitetens skyld.

Et eksempel på en `sources.list` for en http server kan være følgende, hvor vi bruger en dansk ftp-server for de fleste pakker fordi det sparer båndbredde på internettet.

```
deb http://ftp.dk.debian.org/debian/ stable main non-free contrib
deb-src http://ftp.dk.debian.org/debian/ stable main non-free contrib
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb-src http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
```

Ønsker du derimod at tilføje lokale pakkearkiver til din `sources.list`, som for eksempel befinder sig på din harddisk under stien `/var/ftp/debian/dists/woody/main/binary-i386/...`, så tilføjer du

```
deb file:/var/ftp/debian woody main contrib non-free
```

CD kilder tilføjer man ikke manuelt. Man sikrer sig først at CD drevet kan mountes automatisk ved at tilføje linien

```
/dev/hdc /cdrom auto defaults,noauto,ro 0 0
```

til `/etc/fstab` - brug dog dit aktuelle device navn! Derefter tilføjer man Debian CD'er med kommandoen

```
debian:~# apt-cdrom add
```

hvor du indlæser den ene CD efter den anden når du bliver bedt om det. Kig derefter i `sources.list` for at se ændringerne.

Når du har ændret i apts opsætning, så skal du køre **apt-get update** for at opdatere listen over pakker.

Sidder man bag en proxy server kan `apt` også konfigureres til at bruge den. Det sker i `/etc/apt/apt.conf`, som også indeholder anden opsætning af `apt`. Se **man apt.conf** for en komplet reference. For at bruge proxy sættes `Acquire::http::Proxy` til den server, der ønskes brugt. Undtagelser kan sættes ved at sætte variabelen til `DIRECT`:

```
Acquire::http::Proxy "http://wwwproxy:3128/";
Acquire::http::Proxy::ftp.sunsite.dk DIRECT;
Acquire::http::Proxy::mirrors.sunsite.dk DIRECT;
```

Dette sætter dog kun proxy for http, hvis der også skal bruges proxy til ftp, så kan det kun ske ved at sætte `ftp_proxy` i kommandofortolkeren, f.eks.

```
debian:~$ export ftp_proxy=http://wwwproxy:3128/
```

4.2.2. Opgradering af distribution via apt

En af de bedre features ved Debian, er at man kan opgradere sin distribution over internettet, uden overhovedet at genstarte systemet.

Det er sådan, at der findes en `stable` version af Debian, som i skrivende stund hedder `woody` - det er højst sandsynligt den du kører med.

Den næste version af Debian, kommer til at hedde `sarge`, og bliver udgivet engang i fremtiden (når den er klar). Hvad gør du så når den udkommer, og du gerne vil opgradere til den? Du skal ind og redigere i `/etc/apt/sources.list`, her skal du erstatte alle steder hvor der står `stable` eller `woody` til `sarge`. Gem filen, og kør **apt-get update**. Herefter skriver du **apt-get dist-upgrade**.

Nu vil apt-get så fortælle dig hvilke pakker som den har tænkt sig at opgradere, og den vil selv hente pakker hjem, installere dem - og kort tid efter (afhængig af din maskine og din internet forbindelse), vil du have en opgraderet distribution - helt uden at genstarte maskinen.

En anden ting som er meget populær for Debian folk, er at opgradere til de nye versioner (sarge og sid, også kendt som testing og unstable) INDEN de bliver udgivet.

Det smarte ved ovenstående, er at du vil få en Debian maskine som er meget mere "oppe i tiden" end din nuværende woody. Her kan du se, hvad forskellene er ved de tre distributionsversioner:

- woody (stable) - den stabile version af Debian, denne version udgør den nyeste udgave af Debian. Her er de ældste, men mest robuste og gennemtestede pakker.
- sarge (testing) - den næste udgave af Debian, som er under konstant udvikling, sarge erstatter woodys plads som stable i fremtiden. Pakkerne er nyere end i stable, men lidt ældre end unstable.
- sid (unstable) - en version af Debian der aldrig bliver udgivet. Her kommer de nyeste pakker fra Debian udviklerne ind, hvorefter de senere bliver sendt videre til sarge/testing når de er stabile nok.

Når du har opgraderet til en distribution via **apt-get dist-upgrade**, kan du holde den løbende opdateret ved at køre kommandoerne **apt-get update** og **apt-get upgrade**. F.eks. bliver sid/unstable opdateret med nye pakker en gang om dagen.

sid(eller unstable) er endnu en version af Debian som er under udvikling, det er her hvor Debian udviklerne kan lege lidt mere, d.v.s der er ingen garanti for at den virker, men den vil altid indeholde det nyeste software.

De fleste forfattere af denne bog kører selv med enten sarge eller unstable, p.g.a. de nye opdaterede pakker, men hvis du har tænkt dig at opgradere til en af disse versioner, INDEN de bliver released, så skal det nævnes at der er ingen garanti for at det virker - så du skal ikke beklage dig til os :) (de virker dog, 98% af tiden)

4.2.3. Fejlretning og rekonfigurering med dpkg og apt

Når man har problemer med sin Debian box, er det praktisk at kende til nogle af de følgende kommandoer for at danne sig et overblik over situationen. En liste over alle halv installerede pakker og pakker med fejlstatus findes ved

```
debian:~$ dpkg --audit
```

og listen over pakker der er på hold ses ved en af følgende

```
debian:~$ dpkg -l | grep ^h
debian:~$ dpkg --get-selections | grep hold
```

Har man brug for information om hvilken filer en pakke ejer, eller omvendt, hvilken pakke en fil ejes af, kører man

```
debian:~$ dpkg -L pakkenavn
debian:~$ dpkg -S /sti/filnavn
```

Nogle gange er der gået cirkulære afhængigheder i ens system, og så må man gribe til lidt skrappe midler. Følgende simulerer en dist-upgrade hvor både pakker på hold bliver udskiftet med ny, andre pakker, og hvor småfejl i afhængigheder bliver rettet, kommandoen fås med korte eller lange option flags:

```
debian:~# apt-get --fix-broken --show-upgraded --simulate dist-upgrade
debian:~# apt-get -f -u -s dist-upgrade
```

Er man tilfreds med dette, så kører man "for good" med

```
debian:~# apt-get --fix-broken --show-upgraded dist-upgrade
debian:~# apt-get -f -u dist-upgrade
```

Når man afinstallerer en pakke med **apt-get remove pakkenavn**, så forbliver konfigurationsfilerne på computeren, og ved en senere geninstallering med **apt-get install pakkenavn** sparer man tiden for en rekonfigurering (Debian er smart!). Skyldes ens pakke-problemer at nogle pakker ikke er færdig konfigurerede, så trigger man dette ved

```
debian:~# dpkg --configure --pending
```

Ønsker man at rekonfigurere en enkelt pakke, der ellers er vel installeret, bruges

```
debian:~# dpkg-reconfigure pakkenavn
```

Ønsker man derimod at slette en pakke totalt - inklusive dennes konfigurationsfiler - gøres dette ved en af følgende

```
debian:~# dpkg --purge pakkenavn
debian:~# apt-get remove --purge pakkenavn
```

4.2.4. At kun opdatere enkelte pakker

Ofte er det kun enkelte pakker man virkelig har brug for at skulle have i en nyere version. Det kan derfor være smart kun at opdatere de pakker man har brug for at opdatere. (Det kan f. eks. være at du kører woody og vil have fat i en pakke fra sid/unstable).

Først skal man fortælle apt at den skal hente pakkelisterne for både stable, testing og unstable (eller bare to af dem). Det gøres ved at skrive begge distributioner ind i `/etc/apt/sources.list`:

```
deb http://http.us.debian.org/debian stable main contrib non-free
deb http://http.us.debian.org/debian testing main contrib non-free
deb http://http.us.debian.org/debian unstable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb http://non-us.debian.org/debian-non-US testing/non-US main contrib non-free
deb http://non-us.debian.org/debian-non-US unstable/non-US main contrib non-free
```

Dernæst skal pakkelisten opdateres:

```
debian:~# apt-get update
```

Vi skal nu fortælle apt at selv om der findes nyere versioner af en pakke, skal den altid bruge pakken fra stable, med mindre den eksplicit får fortalt andet. Det gøres ved at lave en fil `/etc/apt/preferences`, der indeholder følgende:

```
Package: *
Pin: release a=stable
Pin-Priority: 600
```

Dermed ved apt at den skal give alle pakker, der kommer fra stable, en prioritet på 600. (standard-prioriteten er 500). Når apt skal vælge hvilke pakker den skal installere (opgradere til) vælger den de pakker med højst prioritet (før den ser på versions-numre).

Det er nu muligt at installere enkelte pakker direkte fra testing ved at skrive:

```
debian:~# apt-get install slashem/testing
```

Dette installerer pakken slashem fra testing. Hvis pakken er afhængig af andre pakker, der ikke er i stable, skal disse angives eksplisit. Hvis man ønsker at opfylde alle afhængigheder ved at installere pakker fra testing kan man skrive følgende:

```
debian:~# apt-get --target-release testing install slashem
```

Hvorned apt vil give prioriteten 990 til alle pakker i testing.

Hvis man altid vil følge pakken slashem fra testing, kan man skrive følgende i sin `/etc/apt/preferences`:

```
Package: slashem
Pin: release a=testing
Pin-Priority: 700
```

```
Package: *
Pin: release a=stable
Pin-Priority: 600
```

Så vil slashem fra testing have en højere prioritet en slashem fra stable (600) og slashem fra unstable (500).

Den fulde dokumentation af hvad man kan med `/etc/apt/preferences` findes i manualsiden i `apt_preferences(8)`

4.2.5. Kunsten at holde et woody system opdateret

Debian-brugere løber ofte ind i problemet, at den stabile version af Debian (på nuværende tidspunkt "woody"), er uddateret i forhold til andre distributioner, eller hvad der er behov for. Det er derimod ikke altid nemt at finde den bedste måde at holde sit system opdateret, dette afsnit vil prøve at gøre rede for de foretrukne metoder.

Det er normalt folk der bruger Debian som en arbejdsmaskine (kontra server), der har behov for mere opdateret software. F.eks. er det nyttigt have en nyere version af XFree-serveren hvis ens nykøbte grafikkort kun er understøttet deri, eller måske at få opdateret sin Mozilla til den nyeste og bedste version.

Der er adskillige måder, hvorpå systemet kan holdes opdateret, her er nævnt størstedelen:

- Hente software fra dets hjemmeside og kompilere det manuelt. Dette er ikke anbefalet.
- Fuldstændig droppe woody og opgradere til enten sarge/testing eller sid/unstable (se Afsnit 4.2.2, "Opgradering af distribution via apt" [17]). Dette kan anbefales hvis du ved hvad du gør.
- Hente enkelte src-Debian pakker (pakker med kildekode) fra sarge/testing eller sid/unstable distributionerne, og kompilere dem til din woody boks (se Afsnit 4.3, "Rekompilering af programmer" [20]). Dette er en god mulighed, men ikke altid så lige til.
- Bruge eksternt byggede Debian pakker, som er kompileret til woody med det formål at give folk mulighed for at køre nyere software på woody. Dette er den nemmeste og fortrukne metode. Risikoen er at der ingen garanti er på at disse pakker vil virke efter hensigten.

Problemet med den sidstnævnte metode, er at det ikke altid er lige nemt at finde de pakker man skal bruge. Men der er blevet oprettet et websted, som er dedikeret til formålet, nemlig www.apt-get.org [<http://www.apt-get.org/>]. Her vil du finde (godkendte) kilder som kan bruges til apt med populær software. Kilderne indsættes i `/etc/apt/sources.list`. Af populær software på sitet i skrivende stund, er OpenOffice.org, GNOME 2, KDE, Mozilla, mplayer osv.

4.3. Kompilering af programmer

Selv om mange programmer ligger lige til at hente med **apt-get**, så kan det nogle gange være nødvendigt at

kompile dem selv alligevel. For ikke at ødelægge hele pakkesystemet, så kan det klart anbefales at lave `.deb` pakker og installere dem, i stedet for at kompilere normalt, f. eks. vha. `./configure && make && make install`, som smider en masse filer rundt omkring på diske, som der efterfølgende ikke rigtigt er kontrol med.

Dette afsnit koncentrerer sig om at kompilere pakker, der kan hentes gennem `apt-get`, men som ikke er kompileret helt i forhold til de behov, man selv har. Nogle pakker, der ikke kan hentes gennem `apt-get`, som f.eks. `mplayer` [<http://mplayerhq.hu>] kommer desuden med de nødvendige support-filer til at lave en `.deb` pakke, men for programmer, der ikke har nogle af delene skal der en større proces i gang. Det er alt sammen beskrevet på Debians guide til vedligeholdelse af pakker [<http://www.dk.debian.org/doc/maint-guide/>], og vil ikke blive dækket i dette afsnit.

Processen for at kompilere en pakke er

- Hent kildekoden ned med **`apt-get source`**
- Foretag nødvendige tilpasninger i `debian/rules`
- Hent eventuelle afhængigheder med **`apt-get build-dep`**
- Byg pakken
- Installer pakken

Først er der dog et par supportprogrammer, der skal være installeret. Det drejer sig om pakkerne

- `dh-make`
- `debhelper`
- `fakeroot`
- `debmake`
- `patch`
- `dpkg-dev`

De kan installeres på sædvanlig vis med `apt-get install`. Alternativt kan pakken `build-essential` installeres, den installerer alle pakker, der er nødvendige for at lave `.deb` pakker selv. Derudover skal alle nødvendige programmer til at kompilere med også være installeret. Det drejer sig oftest om følgende:

- `gcc`
- `libc6-dev`
- `cpp`
- `binutils`
- `make`
- `autoconf`
- `automake`
- `perl`

Sidst, men ikke mindst, kræver det også, at der er nogle `deb-src` linier i `/etc/apt/sources.list`. Oftest er det de samme som `deb`-linierne, bare med `deb-src` i stedet for `deb`.

For at illustrere hele processen kommer der et eksempel på hvordan `php4` kompileres.

Først hentes php4 ned:

```
bruger@debian/tmp$ apt-get source php4
Reading Package Lists... Done
Building Dependency Tree... Done
Need to get 3431kB of source archives.
Get:1 http://ftp.sunet.se unstable/main php4 4:4.1.2-1 (dsc) [1527B]
Get:2 http://ftp.sunet.se unstable/main php4 4:4.1.2-1 (tar) [3347kB]
Get:3 http://ftp.sunet.se unstable/main php4 4:4.1.2-1 (diff) [83.0kB]
Fetched 3431kB in 15s (215kB/s)
dpkg-source: extracting php4 in php4-4.1.2
```

Det efterlader php4 i udpakket tilstand i /tmp/php4-4.1.2.

Hvis installationen skal rettes lidt til, så kan det ske nu ved at rette i debian/rules filen, der ligger i den mappe, pakken er pakket ud til. Led efter linier, der starter med configure, og ret configure kommandoen til.

Inden den færdige pakke bygges skal alle de nødvendige afhængigheder hentes ned. Det sker gennem **apt-get build-dep**:

```
debian:/tmp# apt-get build-dep php4
```

Endeligt kan pakken bygges. I dette tilfælde er det faktisk ikke bare en enkelt pakke, men en hel stak, da de enkelte php moduler også bliver bygget som separate pakker:

```
bruger@debian/tmp/php4-4.1.2$ fakeroot
dpkg-buildpackage -b -D
```

Skulle **apt-get build-dep** have overset nogle pakker, så vil buildpackage stoppe med en fejl om hvilke pakker, der mangler, og disse kan så installeres med **apt-get install**. Er alle de nødvendige pakker installeret, så bliver pakken nu bygget. Når den engang er færdig bliver .deb pakkerne lagt i . /, i dette tilfælde altså i /tmp, hvorefter de kan installeres med **dpkg -i** som normalt:

```
bruger@debian/tmp$ ls *.deb
php4-cgi_4.1.2-1_i386.deb      php4-mhash_4.1.2-1_i386.deb
php4-curl_4.1.2-1_i386.deb    php4-mysql_4.1.2-1_i386.deb
php4-dev_4.1.2-1_all.deb      php4-pear_4.1.2-1_all.deb
php4-domxml_4.1.2-1_i386.deb  php4-recode_4.1.2-1_i386.deb
php4-gd_4.1.2-1_i386.deb      php4-snmp_4.1.2-1_i386.deb
php4-ldap_4.1.2-1_i386.deb    php4-sybase_4.1.2-1_i386.deb
php4-ldap_4.1.2-1_i386.deb    php4-xslt_4.1.2-1_i386.deb
php4-mcal_4.1.2-1_i386.deb    php4_4.1.2-1_i386.deb
```

Her kan det selvfølgelig let ske, at nogle af disse pakker er afhængige af andre pakker. I det tilfælde er det letteste normalt at installere den netop byggede pakke med **apt-get**, og derefter installere den version, der lige er blevet bygget, med **dpkg -i**.

Som allerede nævnt, så er der nogle pakker, der ikke ligger til at hente med apt-get, men som er forberedt på at blive lavet som .deb pakke, som f.eks. MPlayer. Her er proceduren den samme som før, det er blot at køre **dpkg-buildpackage -D -b** og installere den .deb pakke, der kommer ud af det.

Pakker man selv har kompileret vil stadig blive erstattet når **apt-get upgrade** køres, og det kan godt være noget irriterende, især hvis pakkerne er kompileret med andre features end standardpakken. Det kan undgås ved at sætte pakken på 'hold':

```
debian:/tmp# dpkg --get-selections|sed 's/\(php4\)([[[:space:]]*)\)\(install\)\/\1\2hold'
```

I dette eksempel angiver strengen 'php4' den pakke, der skal sættes on hold. Når pakker er on hold bliver de ikke længere opgraderet af **apt-get**.

4.4. Problemer med danske specialtegn

Det forekommer tit at Debian ikke er helt 100% klar til at virke med de danske bogstaver, æ, ø og å overalt. Her er beskrevet hvordan du får dem til at virke.

Hvis du har problemer i **console**, så skal du først og fremmest prøve at fjerne udkommenteringen i `/etc/inputrc` ved '**set convert-meta off**'. Udover det kan det anbefales at prøve **dpkg-reconfigure console-data**.

Hvis du har problemer med æ, ø og å i **XFree**, så er der et par ting du kan prøve. Først og fremmest skal der i `/etc/X11/XF86Config(-4)` være refereret til et dansk keyboard i keyboard-sektionen. Dette gør du ved at sætte **XkbLayout** til **dk**.

Hvis nogen GNOME programmer, som f. eks. X-Chat giver dig problemer, så kan det skyldes at du kører et forkert **locale**. Et **locale** er en smart indstilling i Linux der kan ændre sproget på programmer og miljøet på få sekunder.

I Debian kan du sætte localet via **dpkg-reconfigure locales**. Det danske locale hedder `da_DK`.

Kapitel 5. Debian som arbejdsmaskine (desktop)

Debian kan med lidt arbejde hurtigt blive en meget effektiv arbejdsmaskine. Der er masser af valgmuligheder når der skal vælges arbejdsmiljø, Debian indeholder nemlig alle de grafiske miljøer man kan forvente sig af en moderne Linux distribution. Udviklingen indenfor netop dette område sker i et hurtigt tempo, så derfor halter woody (stable-arkivet) lidt bagefter med de nyeste versioner, og af samme grund vælger mange brugere at opgradere til enten testing eller unstable (se Afsnit 4.2.2, "Opgradering af distribution via apt" [17]). Det kan dog anbefales at vente med dette, til man har lidt mere rutine med Debian, se i stedet Afsnit 4.2.5, "Kunsten at holde et woody system opdateret" [20].

5.1. USB-mus

Standardkernerne i Debian woody, 2.2.20 og 2.4.18 - har indbygget USB-understøttelse, men for at USB-understøttelsen kan komme op at køre, så skal man selv indlæse nogle moduler for det. Følgende moduler er oftest brugt:

```
usbcore
usb-ochi, usb-uhci eller uhci (valget af en af disse moduler kommer an på din USB controler)
hid
mousedev
```

Initialiser ovenstående moduler med enten **modconf** (som er et grafisk console program), eller manuelt via **modprobe**:

```
debian:~# modconf
```

Eller:

```
debian:~# modprobe <modulnavn>
debian:~# modprobe usbcore
debian:~# modprobe usb-uhci
debian:~# modprobe hid
debian:~# modprobe mousedev
```

Når USB-musen skal sættes i funktion i X, skal du køre **dpkg-reconfigure xserver-xfree86** - og når du kommer til spørgsmålet om hvor din mus er, skal du vælge `/dev/input/mice` som er USB-devicet på din maskine. Når dette er gjort, skal du genstarte X, og din mus vil forhåbentlig fungere derefter.

Skal du have din mus til at virke i GPM, skal du køre **'gpmconfig'** og sætte driveren til `/dev/input/mice` og mussetypen til 'PS/2'.

En fordel ved USB-mus er at man kan have flere mus tilsluttet samtidig. Desuden kan man tilslutte musen efter computeren er startet op, dette er specielt praktisk til bærbare computere. For at få begge til at virke samtidig i X, skal man definere begge under **Section "ServerLayout"**:

```
InputDevice "USB MOUSE" "CorePointer"
InputDevice "PS2 MOUSE" "Alwayscore"
```

Desuden skal man have hver mus defineret som et inputdevice.

5.2. Lydkort

Lydkort opsætning er relativt nemt - dog kræver det endnu en gang lidt viden om både dit lydkort og rettigheder i Linux.

Det første du skal gøre er at initialisere driveren, eller modulet. Det kræves nu af dig, at du ved hvilket modul du skal indlæse for dit respektive lydkort.

Start **modconf** som root. Her vil du få en oversigt over moduler og deres kategorier. Hvis du kører kerne 2.2.x vil lydkort modulet ligge i kategorien **misc** mens de i kerne 2.4.x vil ligge i kategorien **sound**. Du kan se din nuværende kerne version ved at skrive **uname -r**.

Det gælder så om at finde sit lydkorts modul. f. eks. skal du vælge `emu10k1` hvis du har Soundblaster Live (`emu10k1` er chippen på SB Live). Når du nu indlæser de forskellige moduler, og vil have en anelse om kortet rent faktisk bliver fundet korrekt - så skriv **dmesg** som root.

Når det er gjort, så skal vi ordne rettighederne på `/dev/dsp` og `/dev/mixer` - fordi disse er som standard i Debian, ejet af root - og i gruppen `audio`.

Du skal gøre den bruger, eller de brugere - som skal bruge lydkortet, en del af 'audio' gruppen. Dette gøres ved at udføre følgende kommando:

```
debian:~# adduser foo audio
```

Hvor **foo** er den bruger vi tilføjer. Hvis du vil checke om han blev tilføjet korrekt, kan du kigge i `/etc/group` med din editor.

Bemærk at når du har lavet ændringer i gruppen - så skal brugeren, f. eks. **foo** logge ind igen før systemet opfatter ham som en del af gruppen.

5.3. Grafisk brugergrænseflade (XFree86)

XFree86 er den del af Linux der styrer grafik på din skærm. Versionen der følger med woody er 4.1.0.

Hvis du ikke installerer XFree under installationen, er det nemt at gøre bagefter. Først kan det dog anbefales at installere et par pakker som hjælper XFree installationen med at automatisk finde dit grafikort, mus og skærm:

```
debian:~# apt-get install discover mdetect read-edid
```

Herefter kan du installere X, som vil bruge ovenstående pakker som hjælp til at tilpasse opsætningen efter dit system:

```
debian:~# apt-get install x-window-system
```

Herefter vil alle de nødvendige X pakker blive installeret, forhåbentlig uden problemer. Under installationen af pakken vil et opsætnings program til XFree automatisk starte op. Læs de medfølgende instruktioner, opsætningen er ikke svær.

Når konfigurationen er færdig, kan du skrive **startx** for at starte din ny-konfigurerede x-server. Har du gjort noget galt, kan du bare skrive **dpkg-reconfigure xserver-xfree86** for at køre konfigurations programmet igen.

Hvis du har problemer med X, så se da FAQ'en i Afsnit 8.4, "X og window managers" [81]. Hvis du synes at der mangler noget grafisk på din skærm, så læs videre i Afsnit 5.3.2, "Window managers" [26].

5.3.1. Skrifttyper i XFree86

Fonte i XFree86 er der også altid "problemer" med. I sidste ende handler det om at få installeret de rigtige font pakker. Følgende pakker anbefales fra et dansk synspunkt:

xfonts-base - **Standard font pakken, kræves for at du overhovedet kan starte X.**
xfonts-75dpi - **Et stort sæt af bitmappede fonts i 75dpi, er meget anbefalet.**
xfonts-intl-european - **ekstrem god at have hvis du vil se æ, ø og å i X.**

Hvis du leger meget med grafikprogrammet The Gimp, eller bare skal bruge mange forskellige skrifttyper, så er sharefont og freefont pakkerne måske noget for dig. Disse indeholder en del gode fonte.

Når du skal ud og surfe på det store internet, så er det bedst at få installeret nogen TrueType fonte. Problemet er at de fonte som bliver brugt mest på nettet, er fonte som Arial, Verdana og Tahoma. Disse er lavet af Microsoft, men kan heldigvis hentes på deres hjemmeside.

En Debian pakke kan gøre dette for dig:

```
debian:~# apt-get install msttcorefonts
```

Efter dette skal du indsætte /usr/lib/X11/fonts/TrueType som en FontPath i din /etc/X11/XF86Config(-4).

For at installere andre TrueType fonte, skal du have pakken **fttools** installeret. Dette gøres med **apt-get install fttools**. Når pakkerne er blevet installeret skal du kopiere alle de fonte du ønsker at bruge ind i en mappe, f. eks. /usr/lib/X11/fonts/TrueType.

Når ovenstående er gjort, bruges kommandoen **mktftdir /usr/lib/X11/fonts/TrueType/**. Til sidst skal du tilføje mappen som en FontPath i /etc/X11/XF86Config(-4) hvis du ikke allerede har gjort det. Afslutningsvis skal du genstarte X.

5.3.2. Window managers

XFree86 er kun din grafiske "server", for rent faktisk at få noget grafisk på den - så skal du have en såkaldt window manager. Det er det program der styrer vinduerne. Uden en window manager kommer du ikke langt.

Der findes et væld af window managers, heriblandt blackbox, window maker, sawfish, twm, fvwm, enlightenment m.m.

Installation af en window manager er relativt nemt:

```
debian:~# apt-get install blackbox
```

Dette vil installere blackbox på dit system. Såfremt du har flere window managere inde, og selv vil vælge hvilken der skal være standard når du starter XFree med **startx**, så skal du bruge Debians geniale 'alternatives' system:

```
debian:~# update-alternatives --config x-window-manager
```

Hvis du vil have mere kontrol over, hvad der skal starte med din X, så skal du i din brugers hjemmemappe, oprette filen ~/.xsession (~ er en betegnelse for din brugers hjemme-mappe).

Denne fil beskriver de programmer, der skal starte op. Det er vigtigt at din window manager står til sidst i den. Man kan f. eks. have en .xsession der ser således ud:

```
#!/bin/sh

xclock -bg black -fg green -update 1 -digital -geometry 150x40-0+0 &
xterm -title "Debian GNU/Linux" -ls -geometry 80x24+70+135 &
exec blackbox
```

Der er et mere uddybende eksempel i /usr/share/doc/xfree86-common/examples/.

5.3.3. Installation af KDE

KDE står for K Desktop Environment. KDE er en større mængde programmer som tilsammen skaber en pæn, nydelig og konsistent brugerflade. Et eksempel på en anden brugerflade er GNOME (se Afsnit 5.3.4, "Installation af GNOME" [27]).

KDE 2.2 er inkluderet med Debian woody og kan installeres via apt-get:

```
debian:~# apt-get install kde
```

Efter installationen, skal du lave filen `~/.xsession` (~ er en betegnelse for din brugers hjemme-mappe) med din yndlings editor og indsætte følgende:

```
#!/bin/sh
startkde
```

Herefter kan du starte KDE ved at skrive **startx**.

5.3.4. Installation af GNOME

GNOME er ligesom KDE (se Afsnit 5.3.3, "Installation af KDE" [27]) en større mængde programmer som tilsammen skaber en pæn, nydelig og konsistent brugerflade.

GNOME 1.4 som er inkluderet med Debian woody, har desværre ingen meta-pakke som kan installeres for at få det komplette system, derfor skal man selv definere de enkelte pakker som man vil have gjort til sin GNOME desktop. Her er et godt bud på hvad du har brug for:

```
debian:~# apt-get install gnome-session sawfish-gnome nautilus evolution
```

nautilus er en grafisk filmanager som samtidig giver dig muligheden for at få ikoner på skrivebordet. **evolution** er en kraftfuld e-mail klient. **sawfish-gnome** er en god GNOME window-manager og **gnome-session** er en pakke der afhænger af andre essentielle pakker og får hele din GNOME desktop til at arbejde sammen.

Efter installationen, skal du lave filen `~/.xsession` (~ er en betegnelse for din brugers hjemme-mappe) med din yndlings editor og indsætte følgende:

```
#!/bin/sh
gnome-session
```

Herefter kan du starte GNOME ved at skrive **startx**.

5.3.5. GNOME/GTK+-temaer

En ting, som mange mennesker ikke lige ved, er at man kan bruge temaer på sine GTK+/GNOME programmer. GTK+ er et widget-set og er det som skaber alle knapperne, dropdown boxes, og hvad der nu ellers findes i et moderne grafisk program. Hundredevis af programmer er skrevet i GTK+ og af nævneværdige programmer kan nævnes X-Chat, Gimp, Gabber, gFTP, xmms. Men standard GTK+ tema er ikke et grafisk vidunder, så hvis du vil have noget pænere at kigge på, så læs videre. Det er relativt simpelt at ændre sit GTK+ tema:

Det første du skal gøre, er at installere pakkerne `gtk-engines-pixmap` og `gnome-control-center`:

```
debian:~# apt-get install gnome-control-center gtk-engines-pixmap
```

Herefter vil du være i stand til at starte **gnomecc** (GNOME Control Center), og skifte dit tema.

Der er mange andre pakker med temaer, prøv at udfør **apt-cache search gtk themes**. Også på themes.freshmeat.net [<http://themes.freshmeat.net>] findes der mange gode temaer.

5.4. Drivere til NVIDIA-grafikkort

NVIDIA har lavet deres eget sæt af drivere som kan bruges hvis man vil have 3D-acceleration og andre features ud af sit NVIDIA kort, såsom TV-Out, TwinView m.m. Hvis man ikke umiddelbart har brug for dette, er standard driveren 'nv' fin. For at bruge nyere NVIDIA kort kræves det dog, at du bruger NVIDIAAs egen driver.

Det er nødvendigt at have minimum en 2.4 kerne for at kunne benytte NVIDIA driverne. Det anbefales derfor kraftigt at opgradere kernen til 2.4. Hvis du er i tvivl om du har den rigtige kerne version eller du skal udskifte den nuværende kerne, er der mere information at finde i Afsnit 7.9, "Udskiftning af kernen" [69].

Til at starte med skal du sørge for at have dine kernel-headers installeret. Hvis du selv har kompileret din kerne og smidt den i /usr/src/linux. Så er dette intet problem. Men hvis du bruger en standard Debian kerne, så skal du selv installere kernel-headers pakken, det gøres således:

```
debian:~# apt-get install kernel-headers-$(uname -r)
debian:~# cd /usr/src
debian:/usr/src# ln -sf kernel-headers-$(uname -r) linux
```

Bemærk

\$(uname -r) udskiftes af bash til outputtet af samme kommando

Driverne skal du manuelt hente på NVIDIAAs website. På websitet vil du finde drivere til 3 forskellige slags Linux: Linux IA32, Linux AMD64 og Linux IA64. Du skal højst sandsynligt tage den første - de andre 2 er til 64-bit processorer.

På download-siden skal du først hente driver filen, som i skrivende stund hedder NVIDIA-Linux-x86-1.0-4363.run. Efter filen er hentet, skal du skrive sh NVIDIA-Linux-x86-1.0-4363.run for at eksekvere filen og installere driveren.

Sidste vigtige skridt, er at ændre din /etc/X11/XF86Config-4 så den bruger driveren 'nvidia' i stedet for 'nv'. Derudover skal du lige huske at checke at du har **Load "glx"** der. Husk også at fjerne **Load "dri"** og **Load "GLcore"**, hvis du har disse.

En lille ekstra ting er hvis du f.eks. har forelsket dig i cursor skyggen i Microsoft Windows, kan du snildt opnå samme effekt med dit NVIDIA kort. Du skal blot tilføje et par ekstra linjer under den **Device**-sektion der tilhører dit NVIDIA kort i din XF86Config-4:

```
Option      "SWCursor"          "false"
Option      "HWCursor"          "true"
Option      "CursorShadow"     "true"
Option      "CursorShadowAlpha" "64"
Option      "CursorShadowXOffset" "2"
Option      "CursorShadowYOffset" "1"
```

Bemærk

Denne mulighed er desværre kun tilgængelig på GeForce2 eller nyere grafikkort.

Efter installationen af NVIDIA driverne, vil du have nye muligheder med det medfølgende nvidia-installer program, som skal eksekveres som root.

Tabel 5.1. Parametre til nvidia-installer kommandoen

Parameter	Beskrivelse
<code>--uninstall</code>	Fjerner den nuværende driver. Det fjerner alle filer der tidligere blev installeret på systemet.
<code>--latest</code>	Med dette kan du tjekke om der er kommet en ny

Parameter	Beskrivelse
	udgave af driveren - du skal være forbundet til internettet før dette fungerer.
<code>--update</code>	Hvis en nyere version af driveren af udkommet, kan du hente opdateringen og installere den via denne kommando.

Her er et eksempel på hvordan man henter den nyeste driver:

```
debian:~# nvidia-installer --update
```

5.5. Internet browser

Microsoft har efterhånden fjernet al konkurrence på browserområdet i Windows, ved at inkludere Internet Explorer (IE) som en fast bestanddel. I Windows XP er det dog muligt at afinstallere IE igen, men skaden er sket. De fleste webmastere skriver hjemmesider, som kun bliver testet på IE. Tit er det sådan at disse sider, ikke virker ret godt i andre browsere. Problemet bliver endnu større, hvis siden er skrevet til kun at bruge Windows fonte og siden bliver åbnet i Linux. Heldigvis er det muligt at downloade Windows fonte (se. Afsnit 5.3.1, "Skrifttyper i XFree86" [25]).

Til Linux findes et utal af browsere. I flæng kan nævnes, Mozilla [<http://www.mozilla.org>], Netscape [<http://download.netscape.com>], Galeon [<http://galeon.sourceforge.net/>] og Konqueror [<http://www.konqueror.org>]. Men hvilken browser skal man så vælge? Lad os først se lidt på, hvilke fordele og ulemper der er ved de forskellige browsere. Netscape fandt ud af at de havde tabt browserkrigen til IE, og valgte at lægge sources til browseren ud til Open Source. Ud af dette kom Mozilla. Fordelen ved Mozilla er at bugs forholdsvis hurtigt bliver rettet. Man kan hente et CVS snapshot ned fra dem og derved altid køre med den nyeste. Desuden kommer der med jævne mellemrum nye versioner ud. Da Mozilla er baseret på Netscape er det også nemt at finde plugins til den. Det eneste umiddelbare ulempe er at den er meget tung, og tager lang tid om at starte op. Galeon er baseret på Mozilla kernen, men bruger sin egen grænseflade. Det gør at den ikke føles så tung som Mozilla, men samtidig kan udnytte Mozilla's stærke sider. Konqueror er en browser udviklet af KDE holdet. Konqueror udmærker sig ved at være hurtig, og have samme følelse som IE, men der er sider den er mindre god til at vise.

Hvis du har brug for en console browser, så kan **w3m** eller **lynx** anbefales, begge følger med Debian og kan installeres via apt.

5.5.1. Java og Flash i Mozilla, Galeon og Konqueror

Når man har fået installeret Mozilla, Galeon eller Konqueror, løber man hurtigt ind i det problem at disse ikke kommer med hverken Java eller Macromedia Flash plugins. Heldigvis er dette forholdsvis nemt at installere. For at installere Java 1.3.1 skal man tilføje følgende linje til sin apt source (`/etc/apt/sources.list`):

```
deb ftp://ftp.easynet.be/blackdown/debian woody main non-free
```

Efter en **apt-get update** kan man nu installere Java med **apt-get install j2re1.3**. Man kan tjekke om Java er installeret ved at skrive "about:plugins" i Mozilla-baserede browsere.

Installering af Macromedia Flash er nemt, men manuelt under Woody. Det første du skal gøre, er at downloade `flash_linux.tar.gz` [http://download.macromedia.com/pub/shockwave/flash/english/linux/5.0r50/flash_linux.tar.gz].

Herefter pakker du filen ud, og kopierer dem til dit plugins mappe:

```
bruger@debian~$ mkdir -p ~/.mozilla/plugins/
bruger@debian~$ tar xzfv flash_linux.tar.gz
bruger@debian~$ cd flash_linux/
bruger@debian~/flash_linux$ cp ShockwaveFlash.class libflashplayer.so ~/.mozilla/plugin
```

Genstart Mozilla og/eller Galeon, og herefter skulle Flash være installeret.

Har du et testing eller unstable Debian system kørende, installeres Flash nemmere med **apt-get install flashplugin-nonfree**. Dette kræver at du er forbundet til Internettet, da pakken selv vil prøve at hente Flash fra Macromedias hjemmeside. Genstart Mozilla og/eller Galeon, og herefter skulle Flash være installeret.

Konqueror kræver lidt mere: man skal efter opstart gå i menuerne Settings -> Configure Konqueror -> Netscape Plugins, og aktivere fanebladet "Scan for new plugins". Installation af Flash kan så verificeres ved at skrive "about:konqueror" i browseren. Bemærk at man skal have enten Openmotiv eller LessTif installeret for at Flash plugin virker. Nogle Flash sider er dog programmeret på sådan en måde at man automatisk bliver sendt til Macromedias install side selvom Flash er installeret korrekt. Som regel skyldes det at siderne er programmeret til IE, og derfor ikke "tror" på at Konqueror kan vise Flash. Brug bare linket der sender dig tilbage til Flash siden, og så virker det. Prøv for eksempel www.lego.com/jonnythunder/mummy/ [<http://www.lego.com/jonnythunder/mummy/>]. Husk også at installere pakken **konq-plugins** for andre nyttige KDE plugins med kommandoen **apt-get install konq-plugins**.

5.6. Kørsel af Windows programmer/spil

WINE er et program til Linux der bruges til at køre Windows programmer og spil. WINE står for Wine is Not An Emulator.

For at få det optimale ud af WINE, er det bedst at kompilere selv, specielt hvis du skal have spil i GL mode til at virke optimalt. Derudover er versionen af WINE i Debian woody også gammel i forhold til den man kan hente selv.

Først skal vi hente programmet, dette er en .tar.gz fil. På <http://www.ibiblio.org/pub/Linux/ALPHA/wine/development/> ligger den nyeste version af WINE. Fra dette sted henter du filen med det højeste versions nummer, for eksempel `wine-20020228.tar.gz`. Når du har hentet den nyeste version af WINE ned i din mappe skriver du:

```
bruger@debian:~$ tar xzfv Wine-20020228.tar.gz
```

Så vil **tar** automatisk pakke filerne ud for dig og ligge dem i en mappe der hedder wine-20020228 (hvis dette er versions nummeret). Nu kører vi bare scriptet som ordner det hele for os - `wineinstall` som ligger i `tools` mappen:

```
bruger@debian:~$ cd wine-20020228
bruger@debian:~/wine-20020228$ tools/wineinstall
```

Du vil blive spurgt om adskillige spørgsmål efter programmet er færdigt med at kompilere. Når du er færdig med dette, kan det være at du skal redigere i `~/ .wine/config` før det kører optimalt. Men ellers er det bare at gå ind på sit nye C drev og starte sine programmer:

```
bruger@debian:~/c/windows$ wine notepad.exe
```

5.6.1. Half-Life

Installationen af Half-Life på Linux er nemt, men først skal du installere WINE som beskrevet i Afsnit 5.6, "Kørsel af Windows programmer/spil" [30]. Det første du skal gøre, er at smide din originale Half_life cd i, og så køre `setup.exe` igennem WINE.

```
debian:~# mount /cdrom/
debian:~# logout
bruger@debian:~$ cd /cdrom/
bruger@debian:/cdrom$ wine setup.exe
```

Herefter skal du bare udføre installationen, som hvis det var på Windows. Det kan nu godt være at WINE ikke lukker sig ned efter installationen, men dette kan vi ordne nemt med `killall`

```
debian:~# killall -9 wine
```

Når du så vil starte spillet, skal du gå ind i mappen hvor du har installeret spillet og starte hl.exe med WINE (Her går vi ud fra, at din fake windows mappe ligger i ~/.wine/fake_windows/ og at du har installeret spillet i ~/.wine/fake_windows/Sierra/Half-Life/)

```
bruger@debian:~$ cd ~/.wine/fake_windows/Sierra/Half-Life
bruger@debian:~/.wine/fake_windows/Sierra/Half-Life/$ wine hl.exe -- hl.exe --console
```

Og hvis du har installeret Counter-Strike, skal det åbnes således:

```
bruger@debian:~/.wine/fake_windows/Sierra/Half-Life/ wine hl.exe -- hl.exe --console -g
```

Der kan forekomme fejl på skærmen når du starter Half-Life igennem WINE, men de fleste har ingen effekt.

Et tip kan være, at hvis du gerne vil køre Half-Life i "Fullscreen" skal du trykke Ctrl+Alt+Plus indtil den passer til din opløsning i Half-Life og når du gerne vil ud i din normale opløsning igen skal du trykke Ctrl+Alt+Minus til du når din normale opløsning igen.

I de nyere versioner af CS (1.3 og opefter) kan det give mange problemer hvis man har ingame voice system enabled. Man kan "disable" dette ved at skrive følgende kommandoer i CFG'en eller hvis du allerede har dem kan du udskifte dem med dette:

```
voice_scale "0.000000"
voice_enable "0"
voice_forcemicrorecord "0.000000"
voice_modenable "0"
```

Kapitel 6. Debian som server

Debian er et stærkt server styresystem, og er kendt for blandt andet sin stabilitet og høje sikkerhed. Den høje sikkerheds opnås ved hjælp af regelmæssige sikkerhedsopdateringer fra `security.debian.org`. Debians store pakkearkiv er resultatet af flere års arbejde og test. Det betyder at den software man installerer er utrolig pålidelig. Ulempen ligger i at woodys (stable-arkivet) software kan være af ældre version end hvad man har behov for.

6.1. Webserver

Webserverens primære formål er at servicere filer, som regel HTML dokumenter til klienter via http protokollen - det som vi oplever som 'At surfe på nettet'. Udviklingen af webserveren har gjort at den er blevet mere integreret med andre services som f.eks. databaseservere gennem forskellige kodesprog som også gør dynamisk indhold let at arbejde med.

Men webservere bruges i mange forskellige forbindelser bl. a. når man opdatere sit Debian system med apt-get via internettet (http protokollen), bliver man serviceret af en webserver. De mange forskellige behov har gjort at der er udviklet flere webservere der hverisær forsøger at dække behovene, da det er umuligt at have et produkt som er bedst til det alt.

6.1.1. Installation og opsætning af Apache

Apache webserveren er meget udbredt og er kendt for sin stabilitet, brede funktionalitet og konfigurerbarhed. Linux, Apache, MySQL og PHP danner tilsammen et meget kraftfuldt værktøj til at servicere webapplikationer også kendt som LAMP.

Læs Afsnit 6.2.1, "MySQL" [35] for opsætning af MySQL.

Apache installeres med følgende:

```
debian:~# apt-get install apache
```

Ved at indtaste serverens ip adresse i en internet browser kan man se en HTML testside, såfremt installationen er forløbet uden fejl.

6.1.2. Virtual Hosts med Apache (vhosts)

Virtual Hosts er en feature, der giver dig mulighed for at benytte din Apache webserver med flere værtsnavne (hostnames) og kun en IP. Siden HTTP/1.1 blev indført, har alle browsere sendt en ekstra header med i deres forespørgsler til webservere: **Host**-headeren. Her i angives der det pågældende hostname man forespørger. Derved kan webserveren finde ud af i hvilken mappe, den skal lede efter de forespurgte filer.

Til at starte med skal vi definere hvilken IP og port, der skal lyttes på. Dette klares med kommandoen **NameVirtualHost**. Dette skal være det IP dit netkort har - dvs. at hvis du sidder bag en router, skal du ikke bruge dit globale IP, men derimod det interne. Åbn filen `/etc/apache/httpd.conf` i en editor og tilføj noget lignende:

```
NameVirtualHost 192.168.1.2:80
```

Derudover skal du sørge for, at værtsnavnet svarer til din interne IP fra den box, du vil køre webserver på. Dette kan du let klare i filen `/etc/hosts`. Tilføj følgende i den:

```
192.168.1.2    example.com
192.168.1.2    *.example.com
```


Nu skal vi til at fortælle Apache hvilke værtsnavn vi vil bruge, og hvilke mapper de skal pege på:

```
<VirtualHost 192.168.1.2:80>
DocumentRoot /home/user/public_html/
ServerName example.com
ServerAlias www.example.com
</VirtualHost>
```

DocumentRoot fortæller Apache hvilken mappe der skal danne roden for dette værtsnavn. I dette tilfælde `/home/user/public_html/`. Du skal huske på også at definere adgangsrettigheder til den mappe du vælger. Dette gøres med en **<Directory>**-blok. I vores tilfælde er det ikke et problem da det er defineret i forvejen.

ServerName er det værtsnavn der relaterer til **DocumentRoot**. **ServerAlias** bruges hvis du ønsker mere end et værtsnavn associeret. **ServerAlias** efterfølges af en række mellemrums-separerede værtsnavne. Der kan også laves "stjerne-alias": `*.example.com` (kan dog ikke bruges med **ServerName**).

Skulle du ønske at have en specifik logfil for denne Virtual Host alene kan det ret nemt klares. Eksempel:

```
<VirtualHost 192.168.1.2:80>
DocumentRoot /home/user/public_html/
ServerName example.com
ServerAlias www.example.com
TransferLog /home/user/logs/access.log
ErrorLog /home/user/logs/error.log
</VirtualHost>
```

TransferLog bruges til `access.log`. Mens **ErrorLog** bruges til `error.log` - til at logge serverfejl.

6.1.3. PHP

Apache har et hav af udvidelses moduler. Blandt de mest kendte er skriptsproget PHP. Følgende beskriver hvordan du får PHP til at køre sammen med Apache webserveren.

PHP består af en masse moduler men hovedmodulet ligger i pakken **php4**. Hvis man f. eks. vil integrere PHP med MySQL skal man installere pakken **php4-mysql**.

```
debian:~# apt-get install php4 php4-mysql
```

Dette installerer pakkerne med deres standard opsætning. For at få Apache til at indlæse php, skal følgende linje tilføjes til `/etc/apache/httpd.conf`:

```
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
```

Derefter kører man en **/etc/init.d/apache restart** og hvis det gik godt bør der stå noget i retning af:

```
Apache/1.3.24 (Unix) Debian/GNU PHP/4.1.2 configured -- resuming normal operations
```

```
i /var/log/apache/error.log.
```

Et sidste skridt for at få PHP til at køre, er at man skal fjerne udkommentering (havelågen, #) på følgende linjer i `/etc/apache/httpd.conf`:

```
#AddType application/x-httpd-php .php
#AddType application/x-httpd-php-source .phps
```

Hvis du vil teste om PHP virker på din nye webserver, kan du lave en fil med følgende indhold:

```
<?php phpinfo(); ?>
```

i `/var/www/index.php`. Åbn nu filen i din browser (<http://localhost/index.php>) - du bør nu kunne se at PHP4 kører og også har loadet sit MySQL modul.

6.1.4. Statistik over dine domæner

Webalizer er et statistik program til din Apache webserver. Udfra logfilerne, beregner webalizer besøgsstatistikken for dit website, heriblandt antal besøgende, hvilke browsere der bliver brugt, hvilke tidsrum der er besøgende m.m.

6.1.4.1. Installation og opsætning af Webalizer

```
debian:~# apt-get install webalizer
```

Så skulle webalizer være installeret. Herefter kan du begynde at sætte din webalizer op.

Konfigurationsfilen er som standard `/etc/webalizer.conf` som du skal modificere en lille smule, brug din editor til at redigere den. Se evt. Afsnit 7.4, "Editorer" [59].

Du skal i din `/etc/webalizer.conf` sætte nogle få ting, før det virker. Der er yderligere ting som kan justeres efter behov. Men du skal sætte, **LogFile**, **OutputDir**, **HostName** og du skal hvis du har php sider som index sider på dit website, skal du tilføje **PageType php**. **LogFile** sættes til det dir hvor dine apache log-filer er placeret. Og din **OutputDir** sættes til hvor du vil have dine outputfil placeret f.eks. `/www/domain.dk/webalizer/` og **HostName** sættes til domænet f.eks. `domain.dk`

Du skal også rette lidt i din `/etc/apache/httpd.conf` Det der skal sættes i din `/etc/apache/httpd.conf` er **HostnameLookups** skal sættes til **"On"**, for at webalizer kan identificere hvem brugerne er, laver den så et `hostnamelookup`. **CustomLog** skal sættes til `" /var/log/apache/access.log"`, for at webalizer ved hvilken log fil, den skal beregne statistikken udfra. Nu webalizer være konfigureret til at køre du kan tjekke det ved at køre webalizer.

```
debian:~# webalizer
```

Hvis du skulle udbyde flere domæner og flere brugere skal have adgang til webalizer kan du blot lave en konfigurationsfil for hvert domæne. Hvis du har flere domæner kan du i mappen `/etc/webalizer` ligge konfigurationsfilerne til f.eks. `domain1.dk` og `domain2.dk` og så køre det via kommandoen:

```
debian:~#/etc/cron.daily/webalizer
```

`/etc/cron.daily/webalizer` er et script der gør at webalizer bliver kørt en gang i døgnet. Du kan nu tilgå statistikken via f.eks. `http://domain.dk/webalizer/` hvor du i eksemplet satte `/www/domain.dk/webalizer/` som output dir. Du skulle nu have en fuldt funktionel statistik for dit website.

6.2. Databaseserver

I Debians pakkearkiv kan man finde flere forskellige databaseservere bl. a. MySQL og Postgres. En databaseserver indeholder databaser og dens opgave er at holde styr på hvilke brugere der har adgang til hvilke databaser samt hvilke rettigheder (læse/skrive) brugerne har. Databaser bruges i mange forskellige sammenhænge. Man kan f. eks. bruge en database til at gemme alle sine musik CD'er så man hurtigt kan få en liste frem over dem eller lave søgninger i sin musik samling.

Mulighederne med databaser er mange og i de følgende afsnit beskrives det hvordan man kan installere sin helt egen samt værktøjer til administration.

6.2.1. MySQL

MySQL blev i sin tid populær ved at være lille og hurtig, men har siden udviklet sig meget. På MySQLs hjemmeside kan man finde en præcis beskrivelse af hvilke funktioner de forskellige versioner af MySQL understøtter. Hvis MySQL skal bruges i forbindelse med en applikation, fremgår det som regel af applikationens dokumentation, hvilken version af MySQL der er påkrævet.

Bemærk

Hvis man udvikler eller videredistribuerer et stykke software som ikke er udgivet under en af OSI's godkendte Open Source licenser, hvor MySQL er nødvendig for denne software, skal man have (købe) en kommerciel MySQL licens [<http://www.mysql.com/products/licensing/commercial-license.html>].

Så længe du holder dig til applikationer fra Debians pakkearkiv skulle dette ikke være noget problem.

Fleere applikationer benytter en MySQL database til at gemme deres data i og især webapplikationer som fora, weblogs og ligende. Sådanne applikationer skal som regel kun vide hvilken database de skal bruge og hvilken bruger de kan logge på databaseserveren med, hvorefter de klare resten selv. Derfor vil guiden kun beskæftige sig med nogle grundlæggende ting til at komme hurtig igang. For yderligere dokumentation kan man installere pakken `mysql-doc` eller benytte MySQLs online dokumentation [<http://dev.mysql.com/doc/>].

6.2.1.1. Installation

MySQL installeres med:

```
debian:~# apt-get install mysql-server
```

Når pakken er hentet vil du blive bedt om at svare på forskellige spørgsmål for at sætte MySQL op. Du vil blive spurgt om databaserne skal slettes samtidig med MySQL, hvis man på et tidspunkt skulle vælge at man afinstallere MySQL. Hvis du svare ja skal du huske at tage backup af dine databaser inden du sletter MySQL.

Du skal også fortælle om du vil have at MySQL serveren til at starte ved maskinopstart. For at undgå at skulle starte MySQL manuelt ved et eventuelt reboot, er det en god ide at svare ja til dette. Herefter vil MySQL servicen automatisk blive startet.

6.2.1.2. Sæt administrator kodeord

Efter installation er MySQL-databaseserveren klar til brug, men administratorbrugeren (root) har ikke noget kodeord. Af sikkerhedsmæssige årsager er det en rigtig god idé at give root et kodeord.

Det følgende eksempel vil give root brugeren kodeordet `SiDgAw46`:

```
debian:~# mysqladmin -u root password 'SiDgAw46'
```

Ved at logge ind på databaseserveren får man præsenteret en kommandoprompt, hvor man kan give databaseserveren kommandoer til f. eks. at oprette en ny database og ligge data ind i den. Følgende eksempel logger root brugeren ind på MySQL-databaseserveren (man bliver selvfølgelig bedt om at angive sit kodeord):

```
debian:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 722 to server version: 3.23.49-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> exit
Bye
```

Når man vil logge ud af MySQL kommandoprompten kan man (præcis som i en alm. konsol) skrive **exit** eller bare trykke på Ctrl+D.

6.2.1.3. Opret en database

For at oprette en database skal du logge på databaseserveren med en bruger der har rettigheder til at oprette nye database som f. eks. **root**). Log på som vist i afsnittet fra før og skriv følgende på MySQL kommandoprompten for at oprette en database med navnet **musik**:

```
mysql> CREATE DATABASE musik
```

6.2.1.4. Opret en bruger

Brugerrettigheder i databaseservere kan være lidt langhåret at komme igang med og MySQL er ingen undtagelse. MySQLs brugerrettigheder ligger i database inde i MySQL databaseserveren, så før en bruger kan blive tilføjet skal man først logge ind på serveren og åbne tabellen **mysql**.

```
debian:~# mysql -u root -p mysql
```

Man kan skræddersy rettighederne til sine brugere, hvilket er en rigtig god ide for at opnå den bedste sikkerhed. Men for at alle kan være med, er det følgende eksempel på at oprette en bruger, forholdsvis enkelt. Vi opretter en bruger der har fulde rettigheder til den nye database oprettet i afsnittet Afsnit 6.2.1.3, "Opret en database" [36].

Følgende opretter brugeren **peter** med kodeordet **GoD4wJ?de** og giver ham fuld adgang til databasen **musik**.

```
mysql> GRANT ALL PRIVILEGES ON musik.* TO 'peter'@'localhost'  
mysql> IDENTIFIED BY 'GoD4wJ?de';
```

Bemærk

Denne bruger har ikke adgang til databasen over netværk.

6.2.1.5. Adgang via netværk

Debian's MySQL standardopsætning gør at man ikke kan forbinde sig til databasen via et netværk. Hverken lokalnetværk eller via internettet. Hvis man har brug for at forbinde sig til databaseserveren fra andre maskiner end den maskine som serveren kører på (altså localhost) skal man aktivere det i en opsætningsfil.

Opsætningen foregår i filen `/etc/mysql/my.cnf` og herunder vises den del af filen der har med netværkstilgangen at gøre. **skip-networking** skal være udkommenteret for at man kan få fat i databaseserveren via et netværk.

```
#  
# The skip-networkin option will no longer be set via debconf menu.  
# You have to manually change it if you want networking i.e. the server  
# listening on port 3306. The default is "disable" - for security reasons.  
# skip-networking  
set-variable      = key_buffer=16M  
set-variable      = max_allowed_packet=1M  
set-variable      = thread_stack=128K
```

Når man har foretaget ændinger i opsætningsfilen skal man have MySQL serveren til at genindlæse opsætningsfilen. Dette gøres med:

```
debian:~# /etc/init.d/mysql reload
```

6.2.1.6. Backup af database

Samme med MySQL følger nogle værktøjer til at administrere sin databaseserver. En af værktøjerne bruges til at tage backup af databaser på serveren og hedder **mysqldump**. Følgende eksempel laver en backup, eller et dump som det hedder af databasen **musik**.

```
debian:~# mysqldump -u root -p musik
```

6.2.1.7. Andre brugergrænseflader

Når man installerer MySQL-serveren får man en kommandoprompt som det eneste interface til sin database. At være nødsaget til at skrive kommandoer til sin databaseserver kan hurtigt blive langhåret og ineffektivt. Heldigvis findes der flere alternativer til at administrere sine databaser. Følgende er en liste over Debian pakker med nogle af disse interfaces.

phpmyadmin Hvis man bruger MySQL i forbindelse med webudvikling er det nærliggende at bruge PHPMyAdmin, som er en webgrænseflade til databaseserveren. Man kan også sagtens bruge denne selv om man ikke bruger databasen til webudvikling men man skal være klart over at denne grænseflade er afhængig af en webserver med PHP understøttelse.

Hvis man har mere specifikke behov kan man gennem en webserver med PHP understøttelse få adgang til data i databaserne på database serveren. OpenOffice.org kontorpakken har også nogle indbyggede værktøjer til at integrere databaser. Dette er blot et lille udpluk af mulighederne - der er mange flere.

6.3. FTP-server

Med Debian, følger der en del forskellige FTP-servere, heriblandt kan nævnes **proftpd**, **wu-ftpd**, **pyftpd**, **mudleftpd** og mange flere.

Her vil vi gå i dybden med **proftpd**. proftpd er en FTP-server med mange features, den er meget fleksibel og meget sikker. Når man skal konfigurere den, udnævner den sig specielt ved at dens konfigurations fil har samme format som Apache.

Installation af proftpd er, som med andre pakker, meget nemt:

```
debian:~# apt-get install proftpd
```

Herefter kører din proftpd med det samme. Det er vigtigt at vide, at proftpd bruger systemets bruger som standard - d.v.s. den holder ikke sin egen database over brugere. De brugere som du kan logge ind med i Linux, kan du også logge ind med via FTP. Du tilføjer en ny bruger med kommandoen **adduser**.

6.3.1. Forhindring af at brugere kan se hele fil-hierakiet

Som standard, kan brugere der logger ind via FTP, kigge rundt i hele dit systems fil-hieraki. f. eks.:

```
User foobar logged in.
ncftp /home/foobar > cd /etc
ncftp /etc > get passwd
passwd: 1,05 kB 3,41 MB/s
```

Det er, selvfølgelig, ikke særlig hensigtsmæssigt.

Heldigvis kan proftpd låse brugerne inde i deres hjemme-mapper, således at de ikke kan vandre rundt i hele fil-hierakiet. Dette gøres ved hjælp af konfigurations indstillingen **DefaultRoot**. Følgende linje indsættes derfor i `/etc/proftpd.conf` med din yndlings editor:

```
DefaultRoot ~
```

~ er en tilde og betyder "brugerens hjemme-mappe". Efter du har tilføjet dette i `proftpd.conf`, genstartes proftpd vha. `/etc/init.d/proftpd restart`.

6.3.1.1. At pege på noget ude fra den tillukkede mappe.

Hvis man gerne vil pege på noget ude fra den tillukkede mappe, kan man ikke bare lave en symbolsk lænke, fordi det simpelthen ikke vil virke. Istedet for at bruge symbolske lænker når man skal pege på noget ude fra den tillukkede mappe, kan man bruge "bind"-featuren i programmet "mount". Dette er dog kun muligt i 2.4.x-kerneserien og opefter:

```
debian:~# mount --bind gammelmappe nymappe
```

Dette kræver selvfølgelig at "nymappe" eksisterer, og er en mappe. Så hvis man skal have `/storage/dokumentation/` til at være tilgængelig i `/ftp/dokumentation/`, skriver man denne kommando:

```
debian:~# mount --bind /storage/dokumentation /ftp/dokumentation
```

Man kan selvfølgelig også putte en linie i `/etc/fstab`, så man enten let kan mounte den, ved at skrive "mount `/ftp/dokumentation`", ellers så man kan få systemet til at mounte den ved boot.

```
/storage/dokumentation /ftp/dokumentation ext2 bind,nodev,noexec,noauto 0
```

Man skal selvfølgelig selv rette til, hvis man vil have det på andre måder. Se eventuelt i manpagen for "mount". Hvis man vil have den til at mounte "binden" ved boot, skal man bare fjerne "noauto" og det tilhørende komma foran.

6.3.2. Forhindring af shell-logins til FTP-brugere

Efter du har tilføjet en ny bruger på dit system, har brugeren også adgang til at logge ind via ssh eller telnet hvis disse er installeret på maskinen. Dette er selvfølgelig ikke optimalt, men det kan forhindres.

Det første der skal gøres, er at ændre brugerens shell. Som standard får brugere `/bin/bash` som shell, men ved at ændre dette til `/bin/false` kan de ikke logge ind.

Ændringen af brugeres shell foregår i filen `/etc/passwd`. f. eks. kan en linje i filen se således ud:

```
lise:x:1000:1000:Lise Andersen,,,:/home/lise:/bin/bash
```

Læg specielt mærke til det sidste på linjen, `/bin/bash` - det er dette du skal ændre til `/bin/false`. Desværre får du nu det problem, at proftpd heller ikke vil give brugeren lov til at logge ind. Men dette rettes i `/etc/proftpd.conf` ved at indsætte følgende linje:

```
RequireValidShell off
```

Efter indsættelsen af linjen, skal proftpd genstartes ved hjælp af `/etc/init.d/proftpd restart`.

6.4. DNS-server

DNS står for Domain Name System. Det sørger for at omsætte værtsnavne til ipadresser. f. eks. har **www.jubii.dk** ip-adressen **212.112.128.10**. Hvis man vil finde ip-adressen for et værtsnavn skriver man blot **nslookup**

hostname. Derefter vil din DNS-server (din internet udbyders eller din egen som du vil lære i denne guide at sætte op) spørge en root server, som er en kraftig DNS-server. Denne vil henviser til en topdomain server f. eks. DNS-serverene for **.dk**, som endvidere vil henviser til en DNS-server for f. eks. **jubii.dk**. Endvidere vil jubii's lokale DNS-server sende ip'en for f. eks. **www.jubii.dk** tilbage til dig. Denne process lyder ret tidskrævende, men det sker blot i løbet af nogle få sekunder.

6.4.1. Records

Når man snakker om DNS-server opsætning, skelner man imellem forskellige records:

SOA record	Denne fortæller om hvor langtid recorden kan være i cachen på en caching nameserver (dette begreb bliver forklaret senere). SOA står for Start Of Authority
A record	Denne record henviser direkte til en ip som før omtalt. dvs. at www.jubii.dk har ip adressen 212.112.128.10
AAAA record	Denne record henviser direkte til en ipv6 ip dvs. at www.6bone.net har ipv6 adressen 3FFE:B00:C18:1:0:0:10
CNAME record	Denne record er et alias til et værtsnavn. F. eks. er www.geekworld.dk et alias til geekworld.dyndns.dk
MX record	Denne record står for Mail eXchange record, dvs. den fortæller en mailserver noget om hvilken server der skal modtage mail for det enkelte domain. f. eks. har geekworld.dk 2 mx records, da man ikke kan opnå forbindelse til mailserveren udefra kun fra teledks net, derfor er den primære mx geekworld.dyndns.dk som henviser direkte til serveren, men da der ikke er adgang til den, må man bruge teledanmarks mailgate, derfor er den sekundær, og denne hedder backupmx.post.tele.dk .
NS record	Denne record fortæller hvilken DNS-server der skal bruges til det pågældende domain. f. eks. er ns-recordene til geekworld.dk , ns1.lujer.dk og ns2.lujer.dk , da disse 2 servere har zonefilen til geekworld.dk liggende.

6.4.2. Installation af bind

Bind kan vælges under installationen, hvis man ikke har gjort det kan den installeres med:

```
debian:~# apt-get install bind
```

Så langt så godt, nu ligger bind inde på din computer.

6.4.3. Opsætning af bind

Selve bind opsættes i **/etc/bind/named.conf**. Her er min **named.conf**:

```
options {
    directory "/var/named/zones";
    forwarders {
        194.239.134.83;
    };
};

logging {
    category lame-servers { null; };
    category cname { null; };
};

zone "." {
    type hint;
```

```
        file "/etc/bind/db.root";
};

zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

zone "geekworld.dk" {
    type master;
    file "geekworld.dk";
};
```

Den lange historie, ser sikkert ud for dig som om det er russisk. Men med lidt tålmodighed kan man godt hitte ud af den. Vi starter med at deklarere nogle options nemlig hvor zonefilerne ligger (directory "/var/named/zones");, vi har valgt at ligge vores i /var/named/zones men det er smag og behag. Derefter har vi valgt at bruge Tele Danmarks nameserver som forwarder, dvs. når vi laver et opslag i vores lokale nameserver, så vil vores navneserver automatisk spørge Tele Danmark's nameserver om at søge efter det værtsnavn. Hvis man udelader dette, bruger bind automatisk rootserverne. Derefter bliver der deklareret nogle zones. Det er bl. a. localhost, hvilket vi anbefaler man ligger ind (dette er standard i Debians named.conf). Det spændende kommer når vi deklarerer geekworld.dk. Der fortæller vi bind at zonefilen til geekworld.dk hedder geekworld.dk og som vi allerede har fortalt bind ligger vores zonefiler i /var/named/zones.

6.4.4. Oprettelse af zonefiler

I zonefilen ligger alt information om hvilket værtsnavn der knytter sig til en ip adresse. Her er en af vores zonefiler (geekworld.dk).

```
$TTL 604800
@ IN SOA atlas.geekworld.dk. root.geekworld.dk. (
                                1           ; Serial
                                604800      ; Refresh
                                86400       ; Retry
                                2419200    ; Expire
                                604800 ) ; Negative Cache TTL
;
@ IN NS atlas.geekworld.dk.
@ IN MX 10 atlas.geekworld.dk.
router IN A 192.168.1.1
atlas  IN A 192.168.1.2
venus  IN A 192.168.1.3
apollon IN A 192.168.1.4
```

Det første man støder på er SOA recorden. Den fortæller om levetid, plus at den angiver den primære nameserver (atlas.geekworld.dk) og den angiver desuden email adressen på hostmasteren (root@geekworld.dk). Det næste vi kommer til er NS recorden og MX recorden. Tilsidst angiver vi hvilket ip nummer de forskellige hosts har. Der kan også laves et CNAME, der vil blot stå CNAME i stedet for A, og derefter værtsnavnet, som er aliaset.

Derefter kan vi genstarte bind:

```
debian:~# /etc/init.d/bind restart
```

...og nameserveren vil forhåbentlig virke optimalt. For at bruge nameserveren skal vi have dens ip adresse stående i /etc/resolv.conf:

```
nameserver 192.168.1.1
```

Dette betyder at når vores maskine skal finde ud af IP adressen for et værtsnavn, så spørger vi nameserveren på 192.168.1.1 om IP'en. Dette betyder også at du kan få hjemmelavede værtsnavne for hele dit netværk. Du kan f. eks. oprette dit eget lille intranet hvor <http://familien.jensen/> eksisterer.

6.5. Firewall og deling af internetforbindelse

Fra og med version 2.4 af Linux kernen har Linux brugere fået et stærkt redskab til at gøre internet oplevelsen mere sikker. Dette redskab hedder **iptables**. Hvis du er i tvivl om din kerne er ny nok, eller allerede nu ved at du skal have en nyere kerne installeret, henviser vi til Afsnit 7.9, "Udskiftning af kernen" [69].

I store træk, kan iptables bruges til at dele internettet på et netværk, f.eks. som en gateway til internettet. Iptables kan også bruges til at blokere pakker der passer ind i forskellige regler, populært kaldet en firewall.

Iptables erstatter ipchains som var det foretrukne redskab i tidligere kerne versioner (2.2 og derunder). Den største forskel der er imellem disse to redskaber er, at vi med iptables har fået indført det såkaldte, stateful inspection. Det betyder at vi nu ikke kun kan kigge på pakke typen, afsender og modtager men sågar også status på pakken. Om det er en ny eller igangværende transaktion.

6.5.1. Installation af iptables

Installation af iptables er nemt. Programmet **iptables** hentes via apt-get:

```
debian:~# apt-get install iptables
```

6.5.2. Opsætning af iptables - begreber

Her vil vi forsøge at give et grundlag for at kunne komme igang med iptables, så man kan opsætte og vedligeholde en firewall / router som bruger iptables, samt sikre ens egen computer. Til at starte med gennemgås de forskellige begreber og så gennemgås nogle basis opsætninger man vil komme til at støde på.

Når man snakker iptables er det nogle få vigtige begreber der er vigtige at man kan skelne fra hinanden. Dem der vil blive gennemgået her er chains/kæder, tabeller og policies. En chain kan sammenlignes med en rute. Som standard har man 3 kæder, INPUT, OUTPUT og FORWARD. Når man snakker om chains skal man altid se på det fra Linux kernens synspunkt. Hvis maskinen som kører firewallen, vil sende en pakke, kommer den igennem OUTPUT-kæden, men modtager maskinen derimod en pakke vil kernen kigge om det er en pakke som kommer fordi en bruger som vi er gateway for, har sendt en pakke og bedt om svar. Hvis pakken er svar pakke til en maskine som vi agerer gateway for, vil pakken komme i vores FORWARD chain, ellers vil kernen antage at det er en bruger som har bedt om at snakke med vores server hvor pakken så kommer til vores INPUT chain. Måden at kernen kan se dette på, er ved at kigge på pakken og se om det f. eks. er en SYN (ny pakke) eller en ACK (svar pakke) og matche dette til NAT tabellen hvori informationer om hvem der har sendt requests til hvilke servere ligger så pakker bliver leveret til de rigtige brugere. På denne måde kan vi så vælge at behandle pakken på forskellige måder.

En tabel kan betegnes som en holder til chains. Den eneste tabel vi pt. er interesseret i er faktisk vores nat tabel. Det er nemlig i vores nat tabel at vi skal skrive alle de regler som vi bruger til at lave destinations- eller source NAT (DNAT & SNAT) med. Hvis man f. eks. har mange klienter siddende på et LAN segment og kun en offentlig IP adresse, kan man bruge SNAT til at ændre afsender adressen på vores pakke. Men alt det vil vi komme ind på

senere.

Til sidst har vi så policies. En policy kan bedst betegnes som en standard regel. Hvis vores pakke ikke bliver "fanget" af nogle af vores andre regler i scriptet, vil man kigge på kædens policy. Der er 3 "targets" man kan bruge, nemlig ACCEPT, REJECT og DROP. Hvis man bruger ACCEPT vil pakken blive tilladt at komme igennem, hvorimod DROP simpelthen bare smider pakken væk. Man kan til sidst også bruge REJECT som sender en besked til afsenderen om at denne pakke altså ikke er blevet accepteret. Man vil dog oftest være interesseret i at en maskine fra internettet ikke vil få svar på om du har modtaget pakken eller ej, så derfor bruges DROP oftest fremfor REJECT.

6.5.3. Opsætning af iptables - intro

Iptables er, som næsten alt andet Linux software, case sensitive. Det vil sige at der er forskel på store og små bogstaver, eksempelvis er forward og FORWARD ikke det samme. Det er derfor vigtigt at man sørger for at skrive det rigtigt, da man ellers bare vil få meldt en fejl. Opbygningen af en regel er ikke så syntaks striks som man kunne forvente. Reglen:

```
iptables -A INPUT -p tcp -s 10.20.30.40 -d 89.43.26.32 --dport 80 -j ACCEPT
```

vil f. eks. give det samme resultat som

```
itpables -A FORWARD -s 10.20.30.40 -d 89.43.26.32 -p tcp --dport -j ACCEPT.
```

Tabel 6.1. Iptables basis elementer

Element	Beskrivelse
-s	Source adressen. Bruges til at checke afsender adressen. Dette kan bl.a. bruges hvis man ønsker at en bestemt maskines IP adresse ikke skal have adgang til maskinen. Hvis source adressen angives til FORWARD kæden kan man blokere internet adgangen for en bruger, når ens egen maskine fungerer som gateway / firewall. Man kan i stedet for at angive en IP adresse, kigge på hele subnets ved at skrive følgende: 172.16.0.0/16. Her fungerer vores / som adskillelsen imellem IP og subnet. Tallet efter /et angiver hvor mange subnet bits vi ønsker at checke på. I dette tilfælde er det 16, som giver en subnet maske på 255.255.0.0.
-d	Destinations adressen. Modsvarer -s dog her kigges der på hvem der skal modtage pakken, fremfor hvem der sender pakken. Bruges til at nægte adgang til f. eks. en hjemmeside / server, eller kan bruges til at tillade adgang til en intern server, hvis denne bruges sammen med SNAT.
-p	Protocol. Hvis man ønsker at definere hvilken protokol man vil kigge på, kan vi med -p argumentet vælge imellem f. eks. tcp, udp eller icmp.
--sport & --dport	Source- og destinations port. Hvis vi ønsker at kigge på hvilken port vores trafik kommer fra eller sendes til, kan vi bruge --sport og --dport. Man skal dog, for at kunne bruge disse, også angive hvilken protokol trafikken bruger, med -p argumentet.
-m	Load module. Da iptables er modulbaseret kan man loade flere moduler ved at bruge -m argumentet. Blandt nyttige moduler kan nævnes multiport, mac og state. Alle disse moduler giver så endnu flere muligheder for at checke pakkerne, hvilket er en af forcerne ved iptables, muligheden for skalering.
-t	Table. Denne bruges til at definere hvilken tabel vi ønsker at ændre i.

Disse elementer, eller switches, til iptables er dem som man normalt vil få brug for i den daglige administration af et større eller mindre netværk. Der er dog flere som man typisk ikke vil bruge i et privat hjemmenetværk, f. eks. MAC adresse blokering, men på større kollegie netværk f. eks. kan det oftest være en god metode for at sikre at en bruger ikke kommer på nettet med det samme.

6.5.4. Iptables-eksempler

I denne sektion kan du finde 2 eksempler på brugen af iptables. Det første eksempel bruger en statisk router opsætning, f.eks. en ADSL løsning med en router uden brugen af PPP klienter. Det andet script viser opsætningen med brug af en PPP forbindelse (det kan være alt fra modem, ISDN til PPPoE forbindelse).

Det andet script viser samtidigt hvordan man skal loade iptables, hvis man har valgt at kompilere dette som moduler fremfor en statisk del af kernen.

For at køre scriptsne skal de indsættes i en fil med din yndlings editor hvorefter du skal gøre filen eksekvebar med **chmod +x <filnavn>**. Til sidst køres scriptet ved at skrive **./filnavn**.

6.5.4.1. iptables med router

```
#!/bin/sh

# Variabler som bruges senere i scriptet.
EXT_IP="20.30.40.50"
LAN_NET="192.168.1.0/24"

# Disable routing inden regler påføres
echo 0 > /proc/sys/net/ipv4/ip_forward

# Regler flushes og policies sættes
/sbin/iptables -t nat -F
/sbin/iptables -F
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P INPUT ACCEPT

#####
## BEMÆRK: Denne linie vil give meget output hvis den bruges ##
## og der ikke i /etc/init.d/klogd er ændret KLOGD variabelen ##
## så denne er: KLOGD="-c4" ##
#####
# Logging af nye pakker skal startes
#/sbin/iptables -A FORWARD -m state --state NEW -j LOG --log-prefix IPTABLE_NEW --log-

# Source NAT alle LAN connections til vores externe IP, hvis
# trafikken fra disse ikke er til en af vores LAN IP net.
/sbin/iptables -t nat -A POSTROUTING -s $LAN_NET -d ! $LAN_NET -j SNAT --to $EXT_IP

# Eksempel på at blokkere en enkelt port.
#/sbin/iptables -A FORWARD -p tcp --dport 110 -j DROP

# Eksempel på at blokere for enkelte porte ved brug af multiport modulet. Bemærk
# dog at multiport maksimalt kan tage 15 porte per linie i scriptet.
#/sbin/iptables -A INPUT -p tcp -m multiport --dport 22,53,80,113 -j DROP

# Eksempel på at blokkere porte med en port range
#/sbin/iptables -A FORWARD -p tcp --dport 6666:7000 -j DROP

# Tillad derefter trafik vi selv har sat igang at komme igennem
/sbin/iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Forwarding startes.
echo 1 > /proc/sys/net/ipv4/ip_forward
```

6.5.4.2. iptables med PPP

Scriptet her kan benyttes til en simpel gateway der router alt fra det interne netværk ud til verden. Her bruges 192.168.1.x som ip-adresser på netværket.

```
#!/bin/sh
```

```
# Variabler som bruges senere i scriptet.
LAN_NET="192.168.1.0/24"

# Load de nødvendige moduler hvis iptables ikke er compiled ind i kernen
insmod ip_tables
insmod ip_conntrack
insmod ip_conntrack_ftp
insmod iptable_nat
insmod ipt_MASQUERADE
insmod ip_nat_ftp

# Disable routing inden regler påføres
echo 0 > /proc/sys/net/ipv4/ip_forward

# Regler flushes og policies sættes
/sbin/iptables -t nat -F
/sbin/iptables -F
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P INPUT ACCEPT

# Disse linier masquerader alle LAN connections, og tillader
# alle LAN klienter at kommer igennem firewall'en.
/sbin/iptables -t nat -A POSTROUTING -d ! $LAN_NET -j MASQUERADE
/sbin/iptables -A FORWARD -s $LAN_NET -j ACCEPT
/sbin/iptables -A FORWARD -d $LAN_NET -j ACCEPT
/sbin/iptables -A FORWARD -s ! $LAN_NET -j DROP

# Forwarding startes.
echo 1 > /proc/sys/net/ipv4/ip_forward
```

6.6. DHCP-server

DHCP sørger for at uddele ipadresser på et lokalt netværk - også kaldet dynamisk ip. Det smarte ved dette, er at man bare skal sætte en computer til netværket og så får den automatisk en ipadresse. Det er en kæmpe fordel i sammenligning med at man hver gang skal angive hvilken ip maskinen skal have, og sørge for at ingen andre computere på netværket har samme ip. En DHCP-server virker ved at have et antal ipadresser at dele ud af, og så holder den styr på hvilke ip'er der er delt ud til hvem.

Den DHCP-server vi har valgt at kigge på er udviklet af Internet Software Consortium. Før vi kan installere DHCP-serveren er det bedst at finde ud af hvilket netværkskort DHCP-serveren skal uddele ipadresser på. Dette netværkskort skal være sat op til at have en fast ip, som beskrevet i Opsætning af netværk - "static" ip. For at få en liste over netkort bruges **ifconfig**.

DHCP-serveren installeres ved kommandoen:

```
debian:~# apt-get install dhcp3-server
```

Under installationen vil du blive spurgt om hvilke netværkskort DHCP-serveren skal uddele ipadresser på - den information skulle du gerne have fundet før vha. **ifconfig** - derfor er det bare at skrive netkortets navn ind - f.eks. "**eth0**". Resten af installationen er rent information.

Når installationen er fuldført er det tid til at sætte DHCP-serveren op. Åbn filen `/etc/dhcp3/dhcpd.conf` med din yndlings editor. Denne fil indeholder en masse eksempler på konfigurationer, og det burde være nogenlunde til at finde ud af. Det er smartest at tilføje dine egne subnets sidst i filen. Et eksempel på et subnet kunne være:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.2 192.168.1.254;
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
```

```

option broadcast-address 192.168.1.255;
option domain-name-servers 192.168.1.1;
option domain-name "debianlan";
default-lease-time 600;
max-lease-time 7200;
}

```

Bemærk syntaksen - hver linje afsluttes med et semikolon. Nogle af linjerne kræver nok lidt forklaring:

range 192.168.1.2 192.168.1.254	Denne linje siger at DHCP-serveren skal uddele ipadresser fra 192.168.1.2 til 192.168.1.254. Det er vigtigt ikke at uddele ipadressen for den lokale computer - typisk 192.168.1.1.
option routers 192.168.1.1	Denne linje siger hvilken computer der er gateway på det lokale netværk - opsætning af din computer som gateway dækkes under i Afsnit 6.5, "Firewall og deling af internetforbindelse" [41]
option domain-name-servers 192.168.1.1	Denne linje siger hvilken computer der er DNS-server. Det kan enten være en DNS-server du selv har sat op (som her), eller din internetudbyders DNS-server (Kræver dog at du har sat deling af internetforbindelse op som i Afsnit 6.5, "Firewall og deling af internetforbindelse" [41]). Normalt giver filen <code>/etc/resolv.conf</code> et hint om denne indstilling. Der kan opgives flere DNS-servere ved at bruge komma mellem dem, f.eks. kan TDC kunder skrive option domain-name-servers 193.162.153.164, 194.239.134.83
option domain-name "debianlan"	Denne linje bestemmer domæne navnet for det lokale netværk. Computerne på lokalnettet tror de hedder f.eks. "comp1.debianlan". Linjen kan godt udelades, eller kan sættes til f.eks. "hejmeddig.dk". Pas dog på med at bruge domæne-navne der eksisterer i virkeligheden.
lease-time	De to linjer med "lease-time" siger noget om hvor tit der checkes for om en given ipadresse stadig er i brug, eller om den kan uddeles på ny - standard-indstillingerne kan bruges i stort set alle tilfælde.

Når du er færdig med at rette i konfigurationsfilen er det tid til at genstarte DHCP-serveren.

```
debian:~# /etc/init.d/dhcp3-server restart
```

Hvis alt går som det skal starter DHCP-serveren som den skal, og du kan starte en af de andre computere på netværket, og checke at den får en ipadresse automatisk, hvis den er sat op til DHCP som beskrevet i Afsnit 4.1, "Netværksopsætning" [15]. Hvis der imidlertid opstår problemer vil fejlbeskeder stå i `/var/log/syslog`.

6.7. Mailserver (Courier)

Inden du kaster dig over dette afsnit omkring opsætning af mail-server, bør du vide, at det er et relativt komplekst domæne, som forudsætter, at du i forvejen har en god forståelse for alle de grundlæggende elementer og principper, som et Linux-operativsystem består af og bygger på. Rent videns-/erfaringsmæssigt antages det, at du allerede har læst og forstået indholdet af disse links (eller tilsvarende stof): Linux – Friheden til at lære Unix [<http://www.linuxbog.dk/unix/bog/index.html.php>], Linux – Friheden til systemadministration [<http://www.linuxbog.dk/admin/bog/index.html.php>], Introduction to Linux - A Hands on Guide [<http://www.tldp.org/LDP/intro-linux/html/index.html>], The Linux System Administrator's Guide [<http://www.tldp.org/LDP/sag/html/index.html>].

Courier Mail Server Suite er en komplet mail-server baseret på åbne mail-protokoller som ESMTP, IMAP, POP3, osv. Courier er udviklet og vedligeholdes af Mr. Sam. Courier's web site er www.courier-mta.org [<http://www.courier-mta.org/>].

Courier består af flere individuelle komponenter, herunder SMTP-server, IMAP-server, POP3-server, webmail

server (SqWebMail) og mail-filtrering (maildrop). I det følgende gennemgås opsætning af SMTP-, POP3- og IMAP-server. Ydermere anvendes en database til lagring af brugerdata, så vi kommer ligeledes i let berøring med MySQL-serveren.

Før installation af Courier påbegyndes, er det naturligvis en forudsætning, at ens mail-servers Fully-Qualified Domain Name (FQDN) (på formen <værtsnavn>.<domænenavn>) allerede er at finde i DNS sammen med de nødvendige MX-records osv.

Bemærk, at nogle Internet-udbydere (bl.a. Stofa og TDC) kan have lukket for port 25 i et filter foran deres kunder. Dette bevirker, at mail skal forbi deres server (princippet kaldes 'relay host' / 'smart host'), inden det rammer ens egen server. Se info på disse links: TDC-info [http://privat.tdc.dk/artikel.php?dogtag=tdc_p_int_bre_faq_tek], Stofa-info [<http://www.stofa.dk/showpage.php?conid=95&menuid=118&parentid=4&subparentid=12>]. Dernæst skal du i filen `/etc/courier/esmtproutes` indsætte en linje ned navnet på den server, som er 'relay host'. Har du f.eks. TDC, vil linjen skulle se således ud (bemærk kolonet!):

```
:smtp.mail.dk
```

Forsøg først uden brug af 'relay host', og hvis dette ikke virker, kan du derpå forsøge dig med en 'relay host'.

Hvis en 'relay host' kræver autentifikation, før du kan sende gennem denne, skal du angive de nødvendige oplysninger i filen `/etc/courier/esmtppauthclient`:

```
<relayhost> <brugernavn> <adgangskode>
```

I denne guide bruges 'mailhost.mydomain.org' som FQDN (og IP 62.242.219.250) for vores mail-server. 'mydomain.org' bliver vores første såkaldte virtuelle mail-domæne (for virtuelle mail-brugere oprettet i MySQL-databasen). Mail-serverens egentlige/rigtige systembrugere (herunder root) hører til blandt de ikke-virtuelle brugere, som mail-serveren naturligvis også håndterer mail for.

Verificér for en god ordens skyld, at al DNS ser ud som forventet, inden du fortsætter:

```
mailhost:~# host -t a mailhost.mydomain.org  
mailhost.mydomain.org has address 62.242.219.250
```

```
mailhost:~# host -t a mydomain.org  
mydomain.org has address 62.242.219.250
```

```
mailhost:~# host -t mx mydomain.org  
mydomain.org mail is handled by 10 mailhost.mydomain.org.
```

Med IP 62.242.219.250 og FQDN mailhost.mydomain.org skal `/etc/hosts` se således ud:

```
127.0.0.1          localhost  
62.242.219.250    mailhost.mydomain.org  mailhost
```

Dernæst skal `/etc/hostname` indeholde:

```
mailhost
```

Hvorefter vi lader dette værtsnavn træde i kraft med kommandoen:

```
mailhost:~# hostname --file /etc/hostname
```

Verificér at følgende tre kommandoer giver det ventede uddata:

```
mailhost:~# hostname
mailhost
```

```
mailhost:~# hostname -f
mailhost.mydomain.org
```

```
mailhost:~# dnsdomainname -v
gethostname()=`mailhost`
Resolving `mailhost' ...
Result: h_name=`mailhost.mydomain.org`
Result: h_addr_list=`62.242.219.250`
mydomain.org
```

6.7.1. Installation

Først skal vi have installeret de pågældende pakker. Der er dog en lille hage ved Woody's aldrende version af Courier. Denne tillader nemlig ikke umiddelbart indkomne mails med ugyldige/korruperte MIME headers. "Dette er vel godt?", spørger man så sig selv. Både ja og nej. Ja, fordi standarder nu engang er til for at blive overholdt. Nej, fordi mange mail-klienter (især mange webmail-klienter, bl.a. Hotmail) genererer mails med MIME headers, som bryder med standarderne.

Når Courier så modtager disse mails, bliver de leveret til modtageren med en meddelelse om, at den pågældende mail er korrupert. Den originale mail fra afsenderen er dog velbevaret og vedhæftet som plain text, men det kan undertiden blive en smule irriterende i længden, hvis man modtager mange mails genereret af fejlbehæftede mail-klienter.

Vil man have mulighed for at lade Courier acceptere og videresende mails med korruperte/ugyldige MIME headers, kan man i stedet benytte sig af Courier-pakkerne fra Sarge (testing) eller Sid (unstable). Se eventuelt Afsnit 4.2.2, "Opgradering af distribution via apt" [17] for info om opsætning af apt, så man kan installere pakker fra andet end Woody. I det følgende henter vi pakkerne fra Woody (stable).

Vi starter således med at hente de pågældende pakker:

```
mailhost:~# apt-get install courier-authmysql courier-mta \
courier-imap courier-pop
```

Ud over disse pakker skal man også have en MySQL-server kørende. Se Afsnit 6.2.1, "MySQL" [35] for installation af MySQL samt angivelse af en adgangskode til MySQLs root-bruger, hvilket er særdeles vigtigt! Vælg adgangskoden med omhu.

6.7.2. Opsætning af Courier

Efter pakkerne er blevet installeret på systemet, skal vi have sat Courier op. Der skal redigeres i en del opsætningsfiler, og som en hjælp hertil vises eksempler på, hvorledes disse opsætningsfiler ser ud på et kørende system. Filerne, der skal redigeres i, står skrevet øverst i alle eksempel-filer.

For at Courier kan autentificere brugere via MySQL, skal vi sikre os, at "authmodulelist" indeholder "authmysql" (se eksemplet nedenfor).

Eksempel på /etc/courier/authdaemonrc:

```
authmodulelist="authpam authmysql"
authmodulelistorig="authcustom authcram authuserdb authldap \
```

```
authpgsql authmysql authpam"
daemons=5
version=""
authdaemonvar=/var/run/courier/authdaemon
```

Courier skal kende til oplysninger om MySQL-databasen, som holder på vores brugerdata. Eksemplet nedenfor repræsenterer databasestrukturen, som vi opbygger lidt senere i afsnittet. Vælg en fornuftigt adgangskode (MYSQL_PASSWORD) til din courier MySQL-bruger.

Eksempel på /etc/courier/authmysqlrc:

```
MYSQL_SERVER          localhost
MYSQL_USERNAME        courier
MYSQL_PASSWORD        courier_adgangskode
MYSQL_SOCKET          /var/run/mysql/mysql.sock
MYSQL_PORT            3306
MYSQL_OPT              0
MYSQL_DATABASE        courier
MYSQL_USER_TABLE      users
MYSQL_CLEAR_PWFIELD   password
DEFAULT_DOMAIN        mydomain.org
MYSQL_UID_FIELD       uid
MYSQL_GID_FIELD       gid
MYSQL_LOGIN_FIELD     id
MYSQL_HOME_FIELD      home
```

Sørg for at beskytte denne fil, idet den indeholder en MySQL-adgangskode (defineret ud for MYSQL_PASSWORD):

```
mailhost:~# chmod 600 /etc/courier/authmysqlrc
```

For de flestes vedkommende kræver POP3- og IMAP-delen af Courier ikke det helt store pilleri. Hvis man dog oplever, at ens mail-klient brokker sig over, at alle forbindelser til serveren er opbrugt, er det måske værd at skruer MAXPERIP lidt i vejret i hhv. /etc/courier/pop3d og /etc/courier/imapd, hvis mange brugere kommer fra samme IP-adresse. Hvis du i øvrigt ønsker at beholde mails i din Trash-folder (IMAP) i meget lang tid, bør du også øge værdien (antal dage) ud for IMAP_EMPTYTRASH.

Filen `hosteddomains` indeholder en liste over alle virtuelle mail-domæner på serveren. Hvis man angiver domænenavnene med et punktum foran domænenavnet (f.eks. 'somedomain.org') bevirker dette, at Courier opfatter det som en wildcard-deklaration. Herved opfattes alle sub-domæner af somedomain.org som værende inkluderet i listen over virtuelle mail-domæner. Selve somedomain.org er dog ikke en del af wildcard-deklaration, hvorfor denne også skal listes eksplicit. Vi nøjes i første omgang med blot at smide 'mydomain.org' heri.

Eksempel på /etc/courier/hosteddomains:

```
mydomain.org
```

**** HUSK at køre `makehosteddomains`, hver gang filen er blevet redigeret! ****

I `hosteddomains` har vi netop indtastet vores virtuelle mail-domæner. En anden fil kaldet `locals` har stort set samme virkning som `hosteddomains`, lige bortset fra at domæner listet i `locals` får fjernet deres domænenavn, før modtagerens mailbox slås op af Courier. Eksempelvis vil en mail sendt til `root@mailhost.mydomain.org` blive leveret til root's lokale mailbox i `/root`.

Således bruges `locals` til at angive domæner, der svarer til dine lokale systembrugere, som kan slås op i `/etc/passwd`, mens `hosteddomains` bruges, når du skal have leveret mail til virtuelle brugere i f.eks. en MySQL- eller LDAP-database. Domæner må i øvrigt kun figurere i én af de to filer - aldrig dem begge!

Eksempel på /etc/courier/locals:


```
localhost
mailhost.mydomain.org
```

**** HUSK at køre `makealiases`, hver gang filen er blevet redigeret! ****

Det er altid fornuftig at lade Courier sende al mail til postmaster videre til én selv (f.eks. myself@mydomain.org).

Eksempel på `/etc/courier/aliases/system`:

```
root: postmaster
mailer-daemon: postmaster
MAILER-DAEMON: postmaster
uucp: postmaster
postmaster: myself@mydomain.org
```

**** HUSK at køre `makealiases`, hver gang filen er blevet redigeret! ****

Opret følgende directory:

```
mailhost:~# mkdir /etc/courier/esmtpacceptmailfor.dir
```

Herpå indtastes en liste over samtlige domæner (inkl. localhost), som Courier skal acceptere mails for. Hvis man ønsker det, kan man også heri anvende wildcard-notationen som nævnt i forbindelse med `hosteddomains` -filen (det samme gælder i øvrigt også for `locals` -filen).

Eksempel på `/etc/courier/esmtpacceptmailfor.dir/domains`:

```
localhost
mailhost.mydomain.org
mydomain.org
```

**** HUSK at køre `makeacceptmailfor`, hver gang filen er blevet redigeret! ****

Filen `defaultdomain` må kun indeholde én linje med et domænenavn. Dette domænenavn bruges af Courier i forskellige sammenhænge. Eksempelvis når Courier sender en mail på vegne af en lokal systembruger, fremstår afsenderens mail som lokalbruger@defaultdomain.

Eksempel på `/etc/courier/defaultdomain`:

```
mailhost.mydomain.org
```

Lad Courier acceptere mails med ugyldige/korruperte MIME headers (ikke muligt med Courier fra Woody/stable)

Eksempel på `/etc/courier/bofh`:

```
opt BOFHBADMIME=accept
```

Opret nu en courier-systembruger, som får ejerskab af hele mail-hierarkiet i filsystemet. courier-systembrugeren bruges også til at oprette nye mail-brugeres Maildir-struktur:

```
mailhost:~# groupadd -g 1248 courier
mailhost:~# useradd -m -d /home/courier -g 1248 -u 1248 -s /bin/bash courier
```

Sidst, men ikke mindst, oprettes en Maildir-struktur til **testbruger@mydomain.org**:

```
mailhost:~# su - courier
mailhost:~$ mkdir -p /home/courier/mydomain.org/testbruger
mailhost:~$ mailldirmake /home/courier/mydomain.org/testbruger/Maildir
```

Dette skal du huske altid at gøre i forbindelse med, at du opretter nye mail-brugere på serveren.

6.7.3. Opsætning af MySQL

Vi starter med at oprette en MySQL-database med navn 'courier'. Herpå opretter vi en tabel med navn 'users'. Efter tabellen er oprettet, oprettes en MySQL-bruger med navn 'courier', som får tildelt rettigheder til at kunne bruge den nyoprettede 'courier'-database. Slutteligt indsætter vi brugerdata i tabellen til vores testbruger.

BEMÆRK: Adgangskoden (kaldet 'courier_adgangskode'), som skal indtastes i nedenstående 'GRANT'-kommando, er det samme, som vi tidligere angav i filen /etc/courier/authmysqlrc!

```
mailhost:~# mysql -u root -p
mysql> create database courier;
mysql> use courier;
mysql> CREATE TABLE users (
->   id VARCHAR(40) NOT NULL,
->   uid SMALLINT NOT NULL DEFAULT 1248,
->   gid SMALLINT NOT NULL DEFAULT 1248,
->   password VARCHAR(20) NOT NULL,
->   home VARCHAR(60) NOT NULL,
->   PRIMARY KEY (id)
-> );
mysql> GRANT ALL PRIVILEGES ON courier.* TO courier@localhost
->   IDENTIFIED BY 'courier_adgangskode';
mysql> INSERT INTO users VALUES ('testbruger@mydomain.org', 1248, 1248,
->   'testbruger_adgangskode', '/home/courier/mydomain.org/testbruger');
```

6.7.4. Test af mail-serveren

Nu skulle mail-serveren være klar til brug. En hurtig test med telnet viser, om du kan logge ind via imap/pop3:

```
mailhost:~$ telnet <host> 143
  a login <mail_adresse> <adgangskode>
  a logout
```

```
mailhost:~$ telnet <host> 110
  user <mail_adresse>
  pass <adgangskode>
  quit
```

Prøv også at sende en mail fra en lokal systembruger på serveren til din virtuelle mail-bruger (testbruger@mydomain.org) med kommandoen:

```
mailhost:~$ echo foobar | mail -s test testbruger@mydomain.org
```

Alternativt kan du kommunikere med din smtp-server via telnet. Et eksempel herpå ses nedenfor (serverens output er ikke gengivet i eksemplet), hvor vi sender fra vores testbruger til vores testbruger (bemærk, at vi afslutter vores mail med en linje med kun et punktum):

```
mailhost:~$ telnet <host> 25
  helo localhost
  mail from: <testbruger@mydomain.org>
```

```
rcpt to: <testbruger@mydomain.org>
data
test af mail-server
.
quit
```

Ved samme lejlighed kan du meget passende få bekræftet, at din server ikke er et 'open relay'. Et 'open relay' er en mail-server, som accepterer at sende mail til mail-konti, som ikke tilhører serveren. Efterprøv ovenstående eksempel igen, men udskift denne gang modtager-adressen med en anden (brug et domæne, som ikke tilhører din mail-server). Resultatet skulle meget gerne være, at din server svarer tilbage med 'Relaying denied'. Se <http://www.abuse.net/relay.html> for en mere dybdegående test.

Hvis din smtp/imap/pop3-server mod forventning ikke virker efter hensigten, bør du som det første altid konsultere din `/var/log/mail.log`. Heri er der som regel altid nyttig information at hente.

Hvis alt er i orden, kan du nu oprette din nye mail-konto i din foretrukne mail-klient og derpå sende en mail til dig selv fra kontoen. Du skal bruge følgende oplysninger:

```
Kontotype: IMAP eller POP3 (valgfrit)
Login: testbruger@mydomain.org
Adgangskode: <testbruger_adgangskode>
Server til udgående mail (SMTP): <din ISPs mail-server>
Server til indgående mail (IMAP/POP3): mydomain.org
```

Hvis du ønsker at bruge din egen mail-server til udgående mail, skal du redigere `/etc/courier/esmtpd`. Det drejer sig om aktiveringen af to linjer (`AUTHMODULES="authdaemon"` samt `ESMTPAUTH="LOGIN"`). Du skal cirka 275 linjer ned i filen, før du finder disse linjer (eller blot søg efter 'AUTHMODULES'). Herpå skal du i din mail-klient angive, at du ønsker at bruge din egen smtp-server, og at denne kræver godkendelse (ESMTPAUTH) i forbindelse med forsendelse af mail (brug samme login/adgangskode som ved imap/pop3-login).

Du kan som sagt frit vælge at tilgå din nyoprettede mail-konto som en POP3-service eller som en IMAP-service. Langt de fleste vil nok foretrække IMAP, idet dette giver meget bedre mulighed for at tilgå sin mail-konto fra flere forskellige PC'ere, uden at ens mails bliver spredt for alle vinde. Til forskel fra standard-POP3 forbliver alle mails altid på serveren med en IMAP-konto, og man har mere funktionalitet til rådighed hvad angår mulighederne for at manipulere med ens Inbox. For de nysgerrige, er der følgende link: imap.org/papers/imap.vs.pop.brief.html [<http://imap.org/papers/imap.vs.pop.brief.html>]

6.7.5. Stjernealias

Hvis Courier modtager mails for ikke-eksisterende brugere, vil Courier normalt ikke levere disse, og en mail bliver sendt af sted til afsenderen med en besked om, at den pågældende mail-bruger ikke eksisterer i systemet. Hvis man virkelig ønsker at modtage mails med en ukendt modtager-adresse, kan man lave en 'catch-all'-regel i Courier (dvs. en slags stjernealias).

Bemærk, at en 'catch-all'-regel sandsynligvis fanger mange spam-mails, som du ellers normalt ikke ville have set. Spammere har det med ofte blot at vælge et arbitrært navn foran et domænenavn, og disse mail bliver nu videresendt til en valid mail-adresse. Dette er værd at have med i sine overvejelser, før man sådan uden videre begynder at lave 'catch-all'-regler for all ens domæner.

Der er to måder, hvormed en 'catch-all'-regel kan laves:

Metode 1: 'Cath-all'-regel med maildrop og aliasdir/.courier-default.

```
mailhost:~# apt-get install courier-maildrop
```

Filen `/etc/courier/aliasdir/.courier-default` er det allersidste sted Courier kigger efter instrukser, når det skal afgøres, hvad der skal ske med en mail. Således bruger vi det avancerede mail-filtreringsprogram, maildrop, i denne forbindelse. Kør følgende kommando, som indsætter den fornødne regel:

```
mailhost:~# echo "| /usr/bin/maildrop -d courier" >/etc/courier/aliasdir/.courier-defau
```

maildrop er i øvrigt nødt til at være setuid root i denne sammenhæng. Det gør du med:

```
mailhost:~# chmod u+s /usr/bin/maildrop
```

Herpå laves en maildrop-filtreringsregel i filen `/home/courier/.mailfilter`, som videresender al mail sendt til `*@mydomain.org` til en anden valid bruger i systemet. I eksemplet nedenfor videresender vi til den testbruger, som vi tidligere har oprettet:

```
logfile "/home/courier/maildrop-filter-log"

if ( ( /^To:.*@mydomain.org/ ) || ( /^Cc:.*@mydomain.org/ ) )
{
  to "/home/courier/mydomain.org/testbruger/Maildir"
}

if ( /^To:.* / )
{
  to "/dev/null"
}
```

Al mail sendt til ikke-eksisterende brugere, som ikke falder ind under domænet `'mydomain.org'`, ryger i øvrigt ud i intetheden (`/dev/null`).

Filen `/home/courier/maildrop-filter-log` fortæller, hvilke mails der er processeret via vores regel.

Metode 2: 'Catch-all'-regel med særlig alias-bruger.

Opret en virtuel mail-bruger, kaldet alias, for det pågældende domæne (dvs. `alias@mydomain.org`). Brugeren oprettes på stort set samme vis (minus `maildirmake`, som ikke er nødvendig her), som da `testbruger@mydomain.org` blev oprettet. Samtidig laves en `.courier-default` med instrukser til Courier om at levere til `testbruger@mydomain.org`. Dvs.:

```
mailhost:~# su - courier
mailhost:~$ mkdir -p /home/courier/mydomain.org/alias
mailhost:~$ echo "testbruger@mydomain.org" \
>/home/courier/mydomain.org/alias/.courier-default
mailhost:~$ mysql -u courier -p courier
mysql> INSERT INTO users VALUES ('alias@mydomain.org',1248,1248,
->  ',','/home/courier/mydomain.org/alias');
```

Bemærk, at det er overflødigt/unødvendigt at angive en adgangskode for alias-brugeren. Det er der to grunde til: For det første er det ikke meningen, at man skal kunne logge ind som alias-bruger. For det andet kan det slet ikke lade sig gøre at logge ind som alias-bruger, idet vi bevidst ikke har oprettet nogen Maildir-mappe for alias-brugeren. Således sendes al mail sendt til `*@mydomain.org` nu videre til `testbruger@mydomain.org`. Brugeren `alias@mydomain.org` er blot en slags 'dummy'-bruger, som sørger for, at en 'catch-all'-regel træder i kraft for det pågældende domæne.

Forskellen på de to metoder består i, at metode 1 har en global effekt, mens metode 2 kun har effekt pr. domæne.

6.7.6. Courier Mailing List Manager (couriermlm)

Courier kan håndtere mailinglister via `couriermlm`. `couriermlm` findes som en særskilt Debian-pakke, så lad os starte med at downloade og installere denne:

```
mailhost:~# apt-get install courier-mlm
```

Herpå oprettes mailinglisten, som vi i det følgende eksempel kalder for 'studiegruppe':

```
mailhost:~# su - courier
mailhost:~$ mkdir -p /home/courier/mydomain.org/studiegruppe
mailhost:~$ mysql -u courier -p courier
mysql> INSERT INTO users VALUES ('studiegruppe@mydomain.org',1248,1248,
->  '', '/home/courier/mydomain.org/studiegruppe');
```

Bemærk, at det - ligesom med stjernealiaser - er overflødig/unødvendigt at angive en adgangskode, når du opretter mailinglister i databasen. Ligeledes er det unødvendigt at køre **maildirmake**.

Det antages i det følgende, at du gennem hele processen står i den nyoprettede mappe /home/courier/mydomain.org/studiegruppe. Står du ikke i denne mappe, skal du selv sørge for at tilpasse sti-angivelsen:

```
mailhost:~$ cd /home/courier/mydomain.org/studiegruppe
```

Nu er det tid til at initialisere en mappe, som vi navngiver 'Listdir'. Denne mappe vil holde på alle filer, der har at gøre med vores mailingliste:

```
mailhost:~$ couriermlm create Listdir ADDRESS=studiegruppe@mydomain.org
```

Efter at Listdir-mappen er initialiseret, bør du nok redigere en smule i skabelon-filerne (.tmpl), som dikterer, hvilke beskeder brugere får, når de tilmelder og afmelder sig listen, osv. Der er adskillige skabeloner, og de er ikke overraskende alle på engelsk. Skabelonerne kan (og bør) redigeres, så ordlyden passer til netop din liste. Som et minimum bør du i hvert fald tage et kig på help.tmpl, sub.tmpl og unsub.tmpl.

Nu lader vi Courier vide, at det er couriermlm, som skal stå for at håndtere alle indkomne mails sendt til studiegruppe@mydomain.org:

```
mailhost:~$ echo "| couriermlm msg Listdir" >.courier
```

Angiv nu en mail-adresse på den person, som er ansvarlig for listen:

```
mailhost:~$ echo "mail-adresse" >.courier-owner
```

Nedenstående kommando sørger for, at mails sendt til studiegruppe-<listkommando>@mydomain.org bliver opfattet som kontrolbeskeder. Kontrolbeskeder sendes til couriermlm via specielle mail-adresser. <listkommando> indbefatter bl.a. 'help', 'subscribe' og 'unsubscribe', som henholdsvis anmoder couriermlm om hjælp/information, ønske om tilmelding samt ønske om afmelding:

```
mailhost:~$ echo "| couriermlm ctlmsg Listdir" >.courier-default
```

En liste tildeles forskellige egenskaber med set-kommandoen. Hvis du f.eks. ønsker at blive informeret, hver gang en person tilmelder/afmelder sig listen, kan dette gøres således (se **man couriermlm** for flere muligheder):

```
mailhost:~$ couriermlm set Listdir REPORTADDR=<din_mail-adresse>
```

Udover at folk selv kan tilmelde/afmelde sig listen via hhv. studiegruppe-subscribe@mydomain.org og studiegruppe-unsubscribe@mydomain.org, kan du også gøre dette med følgende kommandoer:

```
mailhost:~$ couriermlm sub Listdir <mail-adresse>
mailhost:~$ couriermlm unsub Listdir <mail-adresse>
```

Ydermere kan du få en oversigt over alle tilmeldte med `lsub`-kommandoen (se **man couriermlm** for flere muligheder):

```
mailhost:~$ couriermlm lsub Listdir
```

Sidst, men ikke mindst, indsættes følgende i courier-systembrugerens crontab (crontab redigeres med kommandoen **crontab -e**):

```
25 * * * * couriermlm hourly /home/courier/mydomain.org/studiegruppe/Listdir
35 4 * * * couriermlm daily /home/courier/mydomain.org/studiegruppe/Listdir
```

couriermlm hourly og **couriermlm daily** udfører hhv. timelig og månedlig vedligeholdelsesarbejde på listen.

En hurtigt test, der gerne skulle vise, at listen nu er tilgængelig, består ganske enkelt i blot at sende en mail til `studiegruppe-help@mydomain.org`. En sådan besked skulle meget gerne få Courier til at returnere en mail med info om listen. Er dette ikke tilfældet, så start med at analysere dine log-filer, for her finder du med al sandsynlighed svar på, hvorfor det gik galt.

Almindelige systembrugere har også mulighed for at oprette egne lister, og idet systembrugere formentligt i forvejen har en `Maildir`-mappe til deres mails, kan man med fordel lave mailinglister som undermapper til `Maildir`-mappen, så man har alle sine mails/maillinglister samlet under én mappe.

Hvis f.eks. brugeren Jens ønsker at oprette en liste kaldet 'kortklub', gøres dette kort fortalt på følgende vis (logget ind som brugeren Jens):

```
mailhost:~$ cd ~
mailhost:~$ couriermlm create Maildir/kortklub ADDRESS=jens-kortklub@mydomain.org
mailhost:~$ echo "| couriermlm msg Maildir/kortklub" >.courier-kortklub
mailhost:~$ echo "jens@mydomain.org" >.courier-kortklub-owner
mailhost:~$ echo "| couriermlm ctlmsg Maildir/kortklub" >.courier-kortklub-default
```

Følgende indsættes i brugerens egen crontab:

```
25 * * * * couriermlm hourly /home/jens/Maildir/kortklub
35 4 * * * couriermlm daily /home/jens/Maildir/kortklub
```

Test at det virker ved at sende en mail til `jens-kortklub-help@mydomain.org`.

6.7.7. Courier Webmail Server (SqWebMail)

Det antages, at du allerede har installeret og konfigureret Apache, så du i forvejen har en funktionsdygtig web-server. Hvis ikke, så slå et smut forbi Afsnit 6.1.1, "Installation og opsætning af Apache" [32].

SqWebMail er Couriers webmail-modul. SqWebMail giver brugere adgang til deres mail via en web-browser på samme vis som med diverse andre webmail-brugergrænseflader (f.eks. den meget populære Squirrelmail). SqWebMail benytter dog ikke IMAP-protokollen, men tilgår i stedet mails på serveren direkte, hvilket giver lynhurtige svartider, når man som bruger manipulerer med sine mails (søger, sletter, flytter, osv.). Bemærk i øvrigt, at SqWebMail kun understøtter `Maildir` [<http://www.courier-mta.org/mbox-vs-maildir/>]-formatet, og altså ikke det traditionelle `Mailbox`-format (mbox).

Start med at installere SqWebMail:

```
mailhost:~# apt-get install sqwebmail
```

Under installationen af SqWebMail vælges blot 'symlink'-metoden.

Sørg for, at du har Apaches CGI-modul aktiveret. Med Apache 1.3.x skal du sikre, at du har en linje i enten

`/etc/apache/modules.conf` eller `/etc/apache/httpd.conf` med `LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so`. Skulle du derimod benytte Apache 2.x, kan du aktivere CGI-modulet på følgende vis:

```
mailhost:~# cd /etc/apache2/mods-enabled && ln -s ../mods-available/cgi.load .
```

Sidst, men ikke mindst, tilføjes en linje med `'Alias /sqwebmail/ /usr/share/sqwebmail/'` til Apaches konfigurationsfil (`/etc/apache/httpd.conf` eller `/etc/apache2/apache2.conf` for hhv. Apache 1.3.x eller Apache 2.x).

Herpå genstartes Apache:

```
mailhost:~# /etc/init.d/[apache|apache2] restart
```

Du kan nu tilgå SqWebMail med en web-browser via adressen `http://mailhost.mydomain.org/cgi-bin/sqwebmail`.

Hvis du synes, denne URL er en smule besværlig at huske/skrive, kan vi med lidt hjælp fra `mod_alias` gøre brug af en linje i Apaches konfigurationsfil, der ser ud som følgende: `'RedirectMatch ^(webmail)$ http://mailhost.mydomain.org/cgi-bin/sqwebmail'`.

Hvis du i grunden ønsker at aktivere muligheden for nemt og smertefrit at lave filtreringsregler for din mail (kræver at maildrop allerede er installeret og konfigureret), behøver du blot gøre følgende:

```
mailhost:~# echo -e "MAILDIRFILTER=../.mailfilter\nMAILDIR=../Maildir" >/etc/courier/ma
```

Nu vil der næste gang, du logger ind via SqWebMail, øverst i skærmbilledet være et link tilgængeligt, kaldet 'Edit Mail Filters'. Herigennem kan du oprette/redigere/slette alskens regler for mail-filtrering.

6.8. Proxyserver

Hvis man har flere maskiner på samme netværk kan det være en fordel at opsætte en proxyserver. En proxyserver kan være med til at sænke belastningen på dit netværk og den kan ligeledes være med til at øge din hastigheden ved almindelig surfen på nettet.

6.8.1. Installation og opsætning af Squid-proxyserver

For at installere squid-proxyserver, kan du med fordel bruge `apt-get`.

```
debian:~# apt-get install squid
```

Dette installerer pakkerne med deres standard opsætning. For at optimere Squid skal du kigge på filen `/etc/squid.conf`

For at komme hurtigt i gang, kan du gå til sektionen i `/etc/squid.conf` hvor der står

```
#  
# INSERT YOUR OWN CONFIGURATION HERE  
#
```

Det gør du med din favorit editor. Hvis du skulle have problemer med din editor, se Afsnit 7.4, "Editorer" [59].

Under ovennævnte afsnit skriver du så for at lave adgang til proxy-serveren via det lokale netværk følgende.

```
acl mynet src 192.168.0.1-192.168.0.20/32  
http_access allow mynet
```

Det ip-interval som der er opstillet skal ses som et eksempel. Eksemplet er vist med ip'erne i intervallet 192.168.0.1 til 192.168.0.20. I din `/etc/squid.conf` skal du også sætte **cache_mem**, **maximum_object_size**, og **cache_mgr**.

```
cache_mem x MB
```

8 MB er standartværdien, du skal være opmærksom på at 8MB ikke er det som selve processen max vil fylde men det som proxy-serveren vil bruge til temp-filer. Selve processen vil ca. fylde 3x det antal MB du stiller tilrådighed.

```
maximum_object_size x KB
```

4096KB er standartværdien, objekter som fylder mere end den satte KB vil ikke blive gemt på proxy-serveren.

```
cache_mgr $USERNAME
```

Her sættes `$USERNAME` til den bruger som skal modtage mails hvis proxy-cachen går ned. Du kan lave bedre og langt mere effektiv opsætning af din proxy, der er meget hjælp at hente på <http://squid.visolve.com/> som er et site, med opsætningsmanual til squid.

Du kan nu starte din squid-proxy server med

```
debian:~# /etc/init.d/squid start
```

Nu er proxyserveren installeret og startet. Squid er en transparent proxy, og klienterne behøver derfor ikke at ændre noget på deres lokale opsætning for at gøre brug af proxyen.

Kapitel 7. Generel Linux

7.1. Basale kommandoer

ls står for at vise en mappes indhold (List directory Contents).

Syntaksen er **ls mappe**, udelades "mappe" listes pågældende mappe.

Eksempler:

```
bruger@debian:~$ ls -l /usr/
```

Dette vil give en lidt mere omfangsrig liste over indholdet.

```
bruger@debian:~$ ls -a /usr/
```

Dette vil vise dig alt hvad der ligger i mappen.

cd står for skift mappe (Change Directory).

Syntaksen er **cd mappe**, udelades "mappe" skiftes til hjemmemappen.

cp står for kopier (Copy).

Syntaksen er **cp fil1 fil2**.

Eksempler:

```
bruger@debian:~$ cp -R ~/dir/ ./dir2/
```

Her kopieres en hel mappes struktur fra /home/user/dir/ til /home/user/dir2/dir/.

mv står for flyt (Move). Kommandoen kan dog også bruges til at omdøbe filer. Hvis man tænker lidt over det er det egentlig et om samme begreb. **mv** fortæller blot fil-strukturen at en mappe eller en fil har ændret navn (evt. også sti). Dette betyder at hvis en fil flyttes lokalt på en harddisk foregår flytningen lynhurtigt fordi den ikke fysisk skal flytte dataene.

Syntaksen er **mv sti1 sti2**. Bemærk at fil1 og fil2 enten kan være en fil eller en mappe.

rm står for slet (Remove). **rm** er derfor en af de mere farlige kommandoer.

Syntaksen er **rm fil**. Fil skal erklæres explicit.

Eksempler:

```
bruger@debian:~$ rm -r dir/
```

Mappen ~/dir/ slettes.

touch står for berør (Touch).

Syntaksen er **touch fil**. Hvis filen eksisterer opdateres tidsstempet på filen. Eksisterer filen ikke oprettes blot en tom fil.

7.2. Pakkede filer (tar, zip)

Tar og gzip/bzip2 er det format man oftest vil se Linux-kildekode distribueret i. Mange gange vil man også støde ind i f.eks. zip filer, her skal du **apt-get install unzip**. For at identificere typen ser man på filens efternavn.

Tabel 7.1. Tabel over pakkesystemer

Filnavn	Kommando til udpakning	Kommando til pakning
*.tar	tar xfv <filnavn>	tar cfv <filnavn> <mappenavn>
*.tar.gz	tar zxfv <filnavn>	tar zcfv <filnavn> <mappenavn>
*.tar.bz2	tar jxfv <filnavn>	tar jcfv <filnavn> <mappenavn>
*.tgz	tar zxfv <filnavn>	tar zcfv <filnavn> <mappenavn>
*.zip	unzip <filnavn>	zip <filnavn> <fil(er) der skal pakkes>

Ovenstående tabel burde give informationer nok til dagligdags brug af disse filtyper. Mere avancerede options findes i "**man tar**" (se afsnittet om man kommandoen)

7.3. Manual sider via man

En af de store fordele ved Linux som operativ system er, at det er fuldstændigt gennem-dokumenteret. Det betyder, at det altid er muligt at finde dokumentation om et emne. Dette afsnit handler om et dokumentations system kaldet **man**, der findes på alle Linux systemer. Kommandoen **man** er grænsefladen til en mængde nyttig information. "man" er en forkortelse for "manual". Næsten alle programmer du har installeret har sin egen såkaldte man-page. Denne side indeholder en beskrivelse af den givne kommando og en gennemgang af brugen og mulige argumenter til den. Derudover indeholder manpage-kartoteket også information om mange andre ting (se kategorier nedenfor) men hovedvægten i dette afsnit vil være på bruger kommandoer. Fordelen (og ulempen) ved denne dokumentations form er, at den er meget kompakt. Man skal nok gå hen og blive forvirret første gang man løber ind i sådan en. Formålet med denne del af vores "lille" Debian intro er at give et overblik over hvordan disse sider skal bruges.

F.eks. for at se kommandoens **ls** manpage skriver du:

```
bruger@debian:~$ man ls
```

Tabel 7.2. De ni kategorier af manpages (taget fra "man man")

Nummer	Beskrivelse
(1)	Executable programs or shell commands
(2)	System calls (functions provided by the kernel)
(3)	Library calls (functions within system libraries)
(4)	Special files (usually found in /dev)
(5)	File formats and conventions (eg /etc/passwd)
(6)	Games
(7)	Macro packages and conventions (eg man(7), groff(7))
(8)	System administration commands (usually only for root)
(9)	Kernel routines [Non standard]

En manpage indeholder flere dele:

Tabel 7.3. Strukturen af en man-page

Sektion	Indhold
NAME	Kommandoens navn og en kort beskrivelse
SYNOPSIS	Indeholder en opsummering af samtlige parametre til kommandoen. Denne del virker ved første øjekast ret voldsom og svær at finde rundt i. Nedenfor er en hurtig forklaring af hvad alle tegnene betyder: fed tekst skriv præcist hvad der står <i>kursiv</i> erstat med passende argument [-abc] en eller flere af argumenterne i [] kan bruges -a -b betyder eller - den ene ELLER den anden, og ikke begge (XOR for programmørerne derude ;) argument ... argument kan gentages [udtryk] hele udtrykket i [] kan gentages f.eks. ses i første linie i SYNOPSIS af "man man": <code>man [-c -w -tZT device]</code> Dette betyder, at kommandoen " man " kan køres med enten parameteren "-c" eller "-w" eller "-tZT", men ikke med flere af dem samtidigt.
DESCRIPTION	Her er en beskrivelse af kommandoen og evt. lidt om dens anvendelse
EXAMPLES	Denne del indeholder som navnet antyder eksempler på brug af kommandoen
OPTIONS	Her er en gennemgang af alle de under SYNOPSIS angivne parametre
FILES	En liste over alle for kommandoen relevante filer
SEE ALSO	Andre manpages med relevant information i forbindelse med denne kommando
BUGS	Ja, hvis der er nogen kendte bugs står de her
AUTHOR	Forfatters/maintainers navn og evt. email-adresse

I øvrigt henvises til man's egen manpage - "**man man**"

7.4. Editorer

Når du arbejder med Debian, kan du næsten ikke undgå at skulle redigere tekstfiler, dette sker både når du skal redigere konfigurations-filer, eller hvis du vil programmere. Dette kapitel prøver at beskrive nogen af de mest kendte editorer, heriblandt kan nævnes:

- vi / vim - en ekstrem god editor, men tager et stykke tid at lære.
- emacs - ekstrem god editor som vi, tager også lang tid at lære.
- nano - en nem editor, men har ikke de samme features som hhv. vi og emacs, dette er den samme editor som Red Hat brugere vil kende som 'pico'.

7.4.1. vi / vim

vi er en af de ældre editorer i UNIX verdenen. Den er tilgængelig på stort set alle UNIX-lignende styresystemer, som f.eks. FreeBSD, OpenBSD, Solaris og selvfølgelig Linux.

Derfor er denne editor ekstremt god at lære, fordi den vil ligge på ethvert system du kommer til at arbejde på. Udover det, så er den smart og effektiv!

Det største problem med **vi**, er at den er svær at lære. Du kan ikke bare starte vi og forvente at du er klar til at redigere filer. Men en ting er sikkert; når du har lært den, og mestrer den, så vil du elske den.

Det er simpelt at starte editoren:

```
bruger@debian:~$ vi
```

Eller også, hvis du vil redigere en eksisterende fil, eller lave en ny:

```
bruger@debian:~$ vi <filnavn>
```

Herfra er du klar til at editere, det første du skal lære, er at der i vi findes 2 forskellige **modes**:

- **Normal mode** - Navigering af filen, små rettelser.
- **Insert mode** - Indsætning/indtastning af tekst.

Du starter normalt i **normal mode**, og for at gå i **insert mode** skal du trykke **i**. Når du har indtastet din tekst, trykker du escape for at gå tilbage til **normal mode**. Og herefter skriver du **:wq** for at gemme og afslutte (write, quit).

Eftersom vi's oprindelige kildekode er copyrightet, så er der lavet mange kloner. Den du normalt vil finde på en Debian box, er klonen der hedder **vim**. vim har en god tutorial hvis du vil lære at blive god til vi.

Du kan starte den "interaktive" gennemgang ved at skrive:

```
debian:~# apt-get install vim
bruger@debian:~$ vimtutor
```

Herved, så brug en 20 minutter på at gå hele gennemgangen igennem, den vil give dig præcise instruktioner, og efter dette vil du føle dig meget bedre tilpas med vi.

Held og lykke.

7.4.2. Emacs

Emacs er ikke kun en editor. Med Emacs får du ikke mindre end hvad man ville kalde "integrated environment"; Du får mail-læser, nyheds-læser, filbrowser og mange andre programmerings-specifikke udvidelser til bla. kompilering og fejlsøgning. Der findes to versioner af emacs. Den oprindelige GNU Emacs. Derudover er der også XEmacs som er optimeret til brug i X. Der er ikke den store forskel rent funktionalitetsmæssigt.

Først skal vi have startet Emacs op:

```
bruger@debian:~$ emacs
```

Hvis du ønsker at benytte dig af XEmacs:

```
bruger@debian:~$ xemacs
```

Selvfølgelig kan Emacs også tage et filnavn som argument.

I Emacs bruger man sekvenser af tastetryk til at udføre kommandoer. **C-x** betyder eksempelvis Ctrl+x. **C-x C-c** vil så være Ctrl+x efterfulgt af Ctrl+c - hvilket giver sekvensen til at lukke Emacs. Derudover er der også en metatast. For det meste er det Alt eller Windowstasten.

Emacs opererer ikke direkte med filerne, men henter dem ind i en såkaldt buffer - i hukommelsen. En buffer behøver ikke kun at kunne indeholde filer. Den kan også indeholde alt muligt andet. En kommandofortolker f.eks. Faktisk kan du have ligeså mange buffere du har lyst til. De tager bare en god portion ram. Du kan få en liste over dine nuværende buffere med følgende sekvens: **C-x C-b**. Hvis du vil hoppe til en af de buffere du har gang i bruger du **C-x b** efterfulgt af navnet på bufferen. Hvis ikke navnet eksisterer oprettes den som en ny buffer. Her kan du selvfølgelig også bruge **Tab** eller **?** for at få Emacs til at hjælpe med at huske navnet.

Når du er færdig med en buffer (evt. gemt filen) kan du lukke bufferen med: **C-x k**.

Du kan også dele Emacs op i flere vinduer. **C-x 2** deler vindue horisontalt. **X-x 3** deler vinduet vertikalt. Hvis du vil tilbage til din udgangs-position bruger du **C-x 1**. For at hoppe mellem vinduerne bruger du **C-x o**.

Det kan være meget brugbart at gemme sine filer undervejs. Dette gøres med sekvensen **C-x C-s**. Hvis du ønsker at gemme under et andet navn bruges sekvensen **C-x C-w**

En alternativ brug af bufferen kunne være at køre en kommandofortolker. Dette gøres meget enkelt vha. sekvensen **M-x shell**.

Emacs har som standard syntax highlighting på mange sprog (også kaldet colorcoding). Den kender forskel på de forskellige formater vha. filens extension ("efternavn"). Men - bla. PHP har den ingen indbygget highlighting på. Så det fixer vi lige med en lille apt-get:

```
debian:~# apt-get install php-elisp
```

Tabel 7.4. Emacs-kommandoer

Sekvens	Kommando
C-x C-c	Afslut Emacs
C-x C-s	Gem
C-x C-w	Gem under nyt navn
C-x C-b	Liste over buffere
C-x b	Hent buffer frem til redigering
C-x k	Dræb den aktuelle buffer
C-x 1	Et vindue (udgangsindstilling)
C-x 2	Del vinduet horisontalt
C-x 3	Del vinduet vertikalt
C-x o	Hop mellem vinduer
M-x shell	Hent en kommandofortolker ind i en buffer

7.4.3. nano (pico)

pico er den editor som mange nybegyndere kender, desværre er den ikke Open Source. nano er en editor som kopierer pico's funktionalitet, og derved bør du altid bruge nano, og det er også den du finder på et Debian system, ikke pico.

Som sagt er nano meget simpel, du åbner en fil således:

```
bruger@debian:~$ nano <filnavn>
```

Efter den er åbnet, vil du se en "menu" nederst i programmet, denne menu viser hvilke valg du har, de står i en form af ^ efterfulgt af et bogstav.

F.eks. betyder **^O**, CTRL + O. Dette vil gemme filen. Ligeledes er **^R** genvejen til at åbne en ny fil. Du afslutter med **^X**.

7.5. Opstartsfiler

I Red Hat og andre distributioner har man en `rc.local` fil til at putte kommandoer ind der skal startes ved opstart af computeren, denne fil findes ikke i Debian. Men fortvivl ikke, Debian har programmer til at håndtere denne problematik.

Hvis du vil have skripts eller programmer til at køre ved opstart af maskinen, så har Debian programmet **update-rc.d**.

Du skal selv lave scriptet med din yndlings editor. Første skridt er at placere dit script i `/etc/init.d`. F.eks. kan vi lave skriptet kaldet `foobar`, det kan se sådan ud:

```
#!/bin/sh

echo "Nu kører foobar scriptet!"
```

Dette script kopierer du til `/etc/init.d/foobar`, hvorefter du eksekverer kommandoen **update-rc.d foobar defaults**. Dette vil generere symbolsk henvisning fra `/etc/rcX.d/foobar` til `/etc/init.d/foobar` - dette sørger for at skriptet bliver startet og evt. afsluttet ved boot/slukning.

Når et script bliver startet, bliver det kørt med **'foobar start'**, og når det skal afsluttes, bliver det kørt med **'foobar stop'**. For at håndtere denne situation, kan vi udvide vores script lidt:

```
#!/bin/sh

case "$1" in
    start)
        echo "nu kører foobar scriptet!"
        ;;
    stop)
        echo "nu stopper foobar scriptet!"
        ;;
    *)
        echo "Usage: /etc/init.d/foobar {start|stop}"
        exit 1
esac
```

Du kan selv prøve at lege rundt ved at køre `/etc/init.d/foobar start` og `/etc/init.d/foobar stop`.

7.6. Fil-rettigheder og Linux' brugersystem

En ting der er svært at forstå som begynder, er Linux' brugersystem og hvordan man sørger for at ens filer har de rigtige rettigheder. Derfor vil vi nu gå i dybden med dette, og forklare det med eksempler der kan lette forståelsen.

7.6.1. Brugeren, gruppen og de andre

Enhver fil på dit Linux system, indeholder informationer om hvem der ejer den, hvilken gruppe den er en del af - og hvad "andre" har af rettigheder til den. Det lyder ret uforståeligt, derfor kigger vi på et konkret eksempel:

```
bruger@debian:~$ ls -l minfil.txt
-rw-r--r--  1 lise      lise           22 21 maj 08:33 minfil.txt
```

Det er nemmest at læse filen bagfra. Filen hedder **minfil.txt**. Filen blev lavet **21. maj kl. 08:33**. Tallet **22** fortæller os at filen indeholder 22 bytes.

Nu er det vigtigt at følge med; de 2 navne **lise** og **lise** er ikke bare en tilfældighed. Det første navn er brugeren som filen er ejet af, det andet navn er gruppen som filen er en del af. Disse navne er vigtige at huske når man skal kigge på det sværeste i linjen - nemlig **-rw-r--r--**.

Det første tegn, **-** fortæller os noget om filtypen, en fil betegnes ved **-** mens en mappe betegnes ved **d** ("directory").

De næste 3 tegn, **rw-** er BRUGERENS rettigheder. D.v.s. **lise** har læse ("**r**", **read**) og skrive ("**w**", **write**) rettigheder. Altså, brugeren **lise** kan både læse fra filen, og skrive til den. Det sidste af de 3 tegn, som er en almindelig streg, er et "felt" der endnu ikke er udfyldt. Det tegn som man kunne have set her er tegnet for eksekverbare rettigheder ("**x**", **execute**). Eftersom der står **-**, betyder det at brugeren lise IKKE har adgang til

at eksekvere filen. Dette betyder ikke så meget, eftersom det ikke giver meget mening at eksekvere en txt (tekst) fil.

De efterfølgende 3 tegn, **r--**, skal læses på samme måde som forklaret før, men dette gælder for GRUPPEN og ikke for brugeren. D.v.s. det gælder i dette tilfælde for gruppen **lise** (du tror måske dette er et brugernavn, men det er det ikke. Der kan være flere brugere med i en gruppe). **r--** betyder her, at gruppen kun har læse rettigheder til filen.

De sidste 3 tegn, **r--**, skal igen læses på samme måde. Men denne gang gælder det for ANDRE. D.v.s. alle andre brugere som ikke er ejeren af filen, eller med i gruppen. F.eks. er brugeren **martin** en af dem der passer ind her. **r--** betyder derfor, at andre brugere kun har læse rettigheder til filen.

Man kan ændre på filers og mappers rettigheder, der er meget fleksibilitet i systemet og samtidig er det overskueligt. Læs næste afsnit for hvordan du ændrer disse rettigheder.

7.6.2. Ændring af rettigheder på filer og mapper

Der er 2 programmer til at ændre rettigheder på filer og mapper, disse er **chmod** og **chown**. Førstnævnte, **chmod**, ("change mode") bruges til at ændre selve rettighederne på filen, d.v.s. de tegn som du læste om i forrige afsnit, f.eks. "**-rw-r--r--**". **chown**, ("change owner"), bruges til at skifte ejeren af filen og gruppen.

Lad os kigge på vores eksempel fra før:

```
bruger@debian:~$ ls -l minfil.txt
-rw-r--r--    1 lise      lise           22 21 maj 08:33 minfil.txt
```

Vi starter med det mest simple program, nemlig **chown**. Forestil dig en situation hvor vil vil have at brugeren **martin** ejer filen i stedet:

```
debian:~# chown martin minfil.txt
```

Bemærk at dette bliver gjort som root. Vi kigger nu på filen igen:

```
bruger@debian:~$ ls -l minfil.txt
-rw-r--r--    1 martin    lise           22 21 maj 08:33 minfil.txt
```

Filen har nu ændret sin bruger til **martin**, det betyder at brugeren som ejede filen før, **lise**, ikke længere kan skrive til filen. Men lise's gruppe er stadig påhæftet filen, med **chown** kan vi ændre brugeren og gruppen samtidig:

```
debian:~# chown martin:martin minfil.txt
```

Efterfølgende vil filen se således ud:

```
bruger@debian:~$ ls -l minfil.txt
-rw-r--r--    1 martin    martin         22 21 maj 08:33 minfil.txt
```

Nu skal vi kigge på et program der er lidt sværere at forstå til at starte med, nemlig **chmod**. **chmod** har forskellige metoder til at ændre rettigheder på. Vi starter med det mest simple og fortsætter med vores eksempel fra før. Lad os antage, at vi vil gøre filen skrivebeskyttet:

```
martin@debian:~$ chmod u-w minfil.txt
-rw-r--r--    1 martin    martin         22 21 maj 08:33 minfil.txt
```

Det første **u** du ser her, betyder **user**. Vi ændrer altså rettighederne for ejeren af filen. **-w** (læses; minus w) fjerner skrive rettighederne på filen. Prøv selv at gør det samme og tjek filens rettigheder efter du har gjort det. **w** vil forsvinde, og du kan ikke længere skrive til filen:

```
martin@debian:~$ echo "foo" > minfil.txt
bash: minfil.txt: Permission denied
```

Her følger et par andre eksempler

Tabel 7.5. Tabel over simple chmod-kommandoer

Kommando	Beskrivelse
<code>chmod a+r <filnavn></code>	Giver læse rettigheder til alle (ejer, gruppe og andre)
<code>chmod u+w <filnavn></code>	Giver skrive rettigheder til ejeren af filen
<code>chmod g+x <filnavn></code>	Giver eksekver-rettigheder til gruppen.
<code>chmod o+r <filnavn></code>	Give læse-rettigheder til andre (o = others).

7.7. Enhedshåndtering

7.7.1. Basal brug af mount

Et af de områder hvor Linux adskiller sig væsentligt fra Windows og DOS er måden hvorpå lagerenheder organiseres. Windows og DOS har et system bestående af drevbogstaver, hvor hvert bogstav er et fysisk drev eller en partition. Således vil A: typisk være en floppydisk og C: almindeligvis den første harddiskpartition.

På Linux er alle lagerenheder bundet sammen i et enkelt hierarki, hvor man i princippet helt selv bestemmer hvor i hierarkiet de enkelte fysiske enheder skal befinde sig. Før en fysisk enhed kan bruges skal den først monteres, til dette bruger man **mount** kommandoen. Hvis man eksempelvis vil se indholdet af en CD der lige er sat i drevet bruges:

```
bruger@debian:~$ mount /cdrom
```

Man kan nu med

```
bruger@debian:~$ ls /cdrom
```

se de filer der ligger på CD'en. Når man er færdig med at bruge CD'en skal den afmonteres igen med:

```
bruger@debian:~$ umount /cdrom
```

Og CD'en kan nu tages ud.

7.7.2. mount og fstab

Mappen /cdrom er en ganske almindelig mappe som enhver anden. Man kan faktisk montere en CD hvor som helst det skal være. Mange Linux distributioner bruger mappen /mnt/cdrom i stedet for.

Når nu det er ligeegyldigt hvor der monteres, hvordan kan systemet så vide at det lige præcis er en CD der skal monteres, og ikke en floppy disk eller en harddisk partition? Svaret findes i filen /etc/fstab. Et eksempel på denne fil ses herunder, men den kan variere meget fra system til system.

```
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/hda1 / ext2 errors=remount-ro 0 1
/dev/hda2 none swap sw 0 0
proc /proc proc defaults 0 0
/dev/fd0 /floppy auto user,noauto 0 0
/dev/cdrom /cdrom auto user,noauto,ro 0 0
```


Hver linie repræsenterer en enhed, som f.eks et CD drev, eller en harddisk partition. De første tre linier - dem som starter med # - er kommentarer, som bliver ignoreret af mount.

Lad os kigge nærmere på en af linierne. Vi bruger den nederste linie, som repræsenterer CD-ROM drevet, til at gennemgå hver enkelt kolonne.

```
# <file system> <mount point> <type> <options> <dump> <pass>
...
/dev/cdrom /cdrom auto user,noauto,ro 0 0
```

Stien /cdrom er angivet i kolonnen **<mount point>**. Det er her programmet **mount** kigger efter når vi fortæller den at der skal monteres noget i den mappe.

I første kolonne ses stien /dev/cdrom, dette er en sti til en særlig fil som repræsenterer det fysiske CD drev.

I tredje kolonne **<type>** angives hvilken type filsystem som skal monteres. Her står **auto**, hvilket betyder at vi stoler på at **mount** selv kan finde ud af hvilket filsystem der ligger på CD'en. Der kunne også være angivet værdien **iso9660** som er den type filsystem som normalt er på en data-CD.

I kolonnen **<options>** angives særlige indstillinger som skal bruges. Den skal vi se nærmere på senere for her kan man angive mange smarte ting. For vores CD drev er der tre indstillinger. **user** betyder at almindelige brugere må montere CD'er, ellers er det normalt kun root der har lov til at bruge **mount** kommandoen. **noauto** angiver at dette drev ikke skal monteres ved opstart. Det er almindeligt at flytbare enheder, som f.eks floppy diske og CD drev ikke monteres ved opstart, i modsætning til harddisk partitioner. Den sidste indstilling **ro** fortæller at enheden skal monteres 'read-only' - altså skrivebeskyttet.

Den sidste kolonne **<pass>** angiver i hvilken rækkefølge diske skal kontrolleres for fejl ved opstart. Her vil der normalt stå 1 for rod filsystemet, 2 for alle andre harddiskpartitioner og 0 for resten, hvilket betyder at de ikke skal kontrolleres.

Hvis ikke der i fstab var en linie for CD drevet kunne man have monteret den alligevel ved at angive en række indstillinger til **mount** kommandoen:

```
debian:~# mount -t auto -o ro /dev/cdrom /cdrom
```

Den opmærksomme læser vil lægge mærke til at hvis man sammenligner ovenstående kommando med indholdet af `fstab`, er det stort set de samme oplysninger, bare i en lidt ændret form. `fstab` gør altså livet en hel del lettere for os som brugere, og sørger for at systemet har de oplysninger det skal bruge til at montere drev ved opstart.

7.7.3. De mest almindelige linier i fstab

CD-ROM eksemplet er ret dækkende for de flestes almindelige behov, de øvrige linier som man typisk støder på gennemgås hurtigt herunder.

```
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/hda1 / ext2 errors=remount-ro 0 1
...
```

Den første linie i vores eksempel der ikke starter med # repræsenterer rod-filsystemet /. Vi ser at den monteres fra partitionen hda1. Filsystemtypen er ext2, som er det mest almindelige filsystem på Linux. Andre muligheder kunne være ext3 eller reiserfs.

Under options ser vi indstillingen **errors=remount-ro**, som blot betyder at filsystemet skal genmonteres skrivebeskyttet hvis der sker fejl.

I dette eksempel er der en stor partition til hele systemet, men man kan også have systemet delt ud på flere partitioner. Hvis man f.eks. har /usr på sin egen partition vil man typisk have en linie som denne:

```
# <file system> <mount point> <type> <options> <dump> <pass>
```

```
...
/dev/hda3      /usr          ext2    defaults          0      2
...
```

Så vil partitionen hda3 som indholder et ext2 filsystem blive monteret under /usr. Indstillingen **defaults** er en nemmere måde at angive en række indstillinger som almindeligvis bruges til faste drev. 2-tallet i slutningen af linien angiver at dette filsystem skal kontrolleres for fejl efter rodfilsystemet. Bruger man et journaliserende filsystem som ext3 eller reiserfs, vil der ikke være behov for at kontrollere for fejl, og man kan da med fordel ændre det til 0.

```
# <file system> <mount point> <type> <options>          <dump> <pass>
...
/dev/hda2      none          swap    sw                0      0
...
```

Anden linie i vores eksempel angiver at /dev/hda2 er en swap partition. Under **Mount point** står der **none**, fordi swap partitioner ikke monteres et bestemt sted som almindelige filsystemer. Med swap partitioner skal der stå **swap** under filsystemtypen og **sw** under indstillinger.

```
# <file system> <mount point> <type> <options>          <dump> <pass>
...
proc           /proc         proc    defaults          0      0
...
```

I tredje linie angives det virtuelle filsystem **proc**. Dette filsystem ligger ikke på en fysisk lagerenhed, men er blot en grænseflade til selve kernen. Det vil komme for vidt her at forklare hvad der ligger i /proc, og hvad det kan bruges til, men linien er nødvendig i fstab.

```
# <file system> <mount point> <type> <options>          <dump> <pass>
...
/dev/fd0       /floppy       auto    user,noauto       0      0
...
```

Fjerde linie er til floppy diske, og minder meget om linien for CD'er. Blot bruges enheden /dev/fd0 og indstillingen **ro** er udeladt fordi floppyer ikke bliver monteret skrivebeskyttet som det er tilfældet med CD'er.

7.7.4. Montering af Windows drev

På Windows bruges der to forskellige typer filsystemer. På Windows 95/98/ME brugtes filsystemet FAT32. På NT, 2000 og XP er filsystemet NTFS standard, dog kan FAT32 også benyttes her. Linux har god understøttelse af FAT32, hvorimod NTFS understøttelse er begrænset. NTFS driveren kan læse uden problemer, men understøttelse af skrivning er endnu ikke færdigudviklet. Derfor bør NTFS partitioner altid monteres skrivebeskyttet.

Windows' filsystemer har en del begrænsninger. Eksempelvis understøtter de ikke Linux' system af rettigheder og filejerskab. Det giver nogle få komplikationer ved montering.

Når et FAT32 drev skal monteres benyttes filsystemtypen **vfat**. Hvis vi antager at der ligger et FAT32 filsystem på hdb1, og vi vil montere det på /mnt/fat kan følgende linie i fstab anvendes:

```
# <file system> <mount point> <type> <options>          <dump> <pass>
...
/dev/hdb1      /mnt/fat      vfat    defaults          0      2
...
```

Forudsat at mappen /mnt/fat findes, kan vores FAT32 filsystem nu monteres som root med kommandoen:

```
debian:~# mount /mnt/fat
```

Hvis vi kigger på filerne på vores fiktive drev, kunne de se nogenlunde sådan ud:

```
drwxr-xr-x root root 4096 Programmer
drwxr-xr-x root root 4096 Dokumenter
-rwxr-xr-x root root 516 autoexec.bat
```

Vi ser at alle filer har samme ejerskab, tilhører samme gruppe, og har samme rettigheder tilknyttet. Det skyldes at disse egenskaber ikke findes på FAT32 drev, og der benyttes derfor de samme standardegenskaber for alle filer. Vi kan dog ændre på hvad der betragtes som standard for hvert filsystem.

Hvis vi vil have at alle filer skal ejes af brugeren anne og tilhøre gruppen fatdrev, kan vi benytte indstillingerne **uid** og **gid**. Det vil se således ud i vores fstab:

```
# <file system> <mount point> <type> <options> <dump> <pass>
...
/dev/hdb1 /mnt/fat vfat defaults,uid=anne,gid=fatdrev 0 2
...
```

Efter montering vil filerne vil nu se ud på denne måde:

```
drwxr-xr-x anne fatdrev 4096 Programmer
drwxr-xr-x anne fatdrev 4096 Dokumenter
-rwxr-xr-x anne fatdrev 516 autoexec.bat
```

Tilladelserne kan også ændres med indstillingen **umask**. **umask** bruger numeriske værdier for tilladelser ligesom kommandoen chmod. Dog fungerer det lidt anderledes da **umask** angiver hvilke tilladelser der ikke skal være til stede. Hvis vi vil have at ejeren skal have alle rettigheder, at gruppen skal kunne læse og eksekvere og andre ingen rettigheder skal have (rwxr-x---) bruges **umask** værdien 027. (0 betyder alle rettigheder, 2 betyder alle rettigheder bortset fra skrivning og 7 betyder ingen rettigheder). Dette kan vi tilføje til vores fstab:

```
# <file system> <mount point> <type> <options> <dump>
...
/dev/hdb1 /mnt/fat vfat defaults,uid=anne,gid=fatdrev,umask=027 0
...
```

Og nu vil filerne se sådan ud:

```
drwxr-x--- anne fatdrev 4096 Programmer
drwxr-x--- anne fatdrev 4096 Dokumenter
-rwxr-x--- anne fatdrev 516 autoexec.bat
```

Ved montering af NTFS filsystemer bruges typen ntfs, og man bør altid montere dem skrivebeskyttet. Her ses et eksempel på dette:

```
# <file system> <mount point> <type> <options> <dump>
...
/dev/hdb2 /mnt/ntfs ntfs defaults,ro 0
...
```

NTFS driveren bruger samme metode til angivelse af ejerskab og tilladelser som FAT32

Udover de nævnte indstillinger kan man i **man 8 mount** se hvilke indstillingsmuligheder der er for hvert enkelt filsystem.

7.7.5. Montering af disk-aftryk

Når man henter eksempelvis en Linux CD fra internettet, vil man typisk få en iso fil, som man derefter normalt vil

brænde til en CD for at kunne bruge den. Det er dog også muligt at montere en sådan fil direkte, så man kan se hvad der er på inden man brænder den. Kommandoen til at gøre dette kunne se ud på denne måde:

```
debian:~# mount -t auto -o ro,loop debian-30r5-i386-binary-1.iso /mnt/isocd
```

Første argument **-t auto** angiver filsystemtypen, her bruger vi **auto** som med en almindelig CD.

Andet argument **-o ro,loop** angiver at vi vil montere som skrivebeskyttet ligesom med normale CD'er, og at vi benytter det der hedder en loopback enhed til at montere.

Derefter kommer filnavnet til iso filen, og den mappe hvor filsystemet skal monteres. Som monteringspunkt bruges her stien `/mnt/isocd`, men der er helt frit valg, man skal blot huske på at mappen skal være oprettet i forvejen.

7.8. At finde rundt på et Linux system

Linux' filsystem kan være svært at overskue når man har været vant til strukturen i f.eks. Windows. Men fortvivl ikke; der er logik bag det hele.

Noget som Debian udemærker sig i, er at der er god kontrol over placeringer af filer, fordi Debian følger FHS [<http://www.pathname.com/fhs/>] (Filsystem Hieraki Standarden). Derfor vil man meget sjældent opleve at en fil ikke er placeret korrekt.

Linux har ikke på samme måde som Windows, drev betegnet som bogstaver (f.eks. C: og D:). Linux' filsystem starter i det man kalder roden og det betegnes med en enkelt skrånstreg ("/"). Fra roden, bygges fil-systemet logisk op. Her er en oversigt over de mest nyttige mapper at kende til:

Tabel 7.6. Oversigt over Linux' fil-hieraki

Sti	Beskrivelse
/	Rod-mappen. Det er her hvor hele træet starter.
/bin	En mappe der indeholder de mest basale programmer (bin betyder "binary", d.v.s. binære filer)
/boot	Indeholder filer til boot-loaderen (f.eks. LILO)
/dev	Specielle enheds-filer, som referer til fysiske enheder
/etc	Indeholder konfigurations filer som gælder for hele det lokale system.
/home	Indeholder hjemme-mapper for alle brugerne på systemet. (Undtagen root, som har sit i /root)
/lib	Indeholder biblioteker som er krævet af programmer der bruges til at starte systemet.
/proc	proc fil-systemet er et sted hvor kernen registrerer en del informationer om systemets tilstand.
/root	Root-brugerens hjemme-mappe.
/sbin	Ens med /bin, men indeholder programmer som normalt kun eksekveres som root.
/tmp	Står for temporary (midlertidig), indeholder tilfældige filer som programmer laver midlertidigt.
/usr	Denne indeholder en hel del; blandt andet størstedelen af programmerne på systemet, bibiloteker, delte filer og dokumentation.

Yderligere dokumentation for disse og flere mapper kan ses i man-siden [hier\(7\)](#).

7.8.1. Søgning efter filer på systemet

Ofte vil man gerne finde ud af hvor en fil er placeret på systemet. Der er en del forskellige måder at gøre det på. Hvilken metode der er den optimale, kommer an på hvad du søger efter.

En metode som altid vil virke, er at bruge **find** kommandoen. Ulempen er at den er langsom i forhold til **locate** som vi fortæller om senere. Fordelen er at den er ret fleksibel og giver mange muligheder i søgningen. Den bruges således:

```
bruger@debian:~$ find / -name '*.jpg'
/home/lise/billedel.jpg
/usr/share/Eterm/pix/tile/blackstone.jpg
[...]
```

Her angiver vi først at vi vil starte søgningen i roden ("/"). Her kunne vi også have angivet punktum (".") for at bare søge fra den nuværende mappe. Som det andet argument angiver vi at vi vil have alle filer der har en endelse på .jpg, her kunne vi også sagtens have skrevet et præcist filnavn hvis vi kun søgte efter en specifik fil.

En hurtigere metode end **find** er **locate** kommandoen. **locate** har den fordel, at den er meget hurtig - fordi den kigger i en database over alle filer på systemet. **locate** kører automatisk om natten og opdaterer sin database, så den altid er up-to-date. Men det betyder så også, at man ikke kan regne med at finde helt ny-oprettede filer medmindre man opdaterer databasen manuelt. Kommandoen bruges således:

```
bruger@debian:~$ locate mintekstfil.txt
/home/lise/mintekstfil.txt
```

Hvis man vil opdatere databasen manuelt, skal man køre kommandoen **updatedb** som root.

Mange gange kan det være at man bare søger efter et program på systemet. Eftersom der er styr på hvor programmer og dets tilhørende komponenter ligger, er det nemmere bare at søge i disse. Derfor har man **whereis** kommandoen. Ulempen er at den ikke søger i hele filsystemet. Den bruges således:

```
bruger@debian:~$ whereis vi
vi: /bin/vi /usr/bin/vi /usr/share/man/man1/vi.1.gz
```

7.9. Udskiftning af kernen

Man kan hurtigt komme ud for at det er nødvendigt at udskifte ens kerne, dette kan f.eks. være pga. fejlrettelser, nye features, ny understøttelse af hardware eller fordi ens nuværende kerne indeholder et kendt sikkerhedshul. Måske er man bare ude efter en optimeret kerne tilpasset ens eget system.

Sammen med en installation af Woody følger en 2.2.20 eller en 2.4.18 kerne som standard. Hvis du ikke ved eller ikke kan huske hvilken kerne du har kan kommandoen **uname** hjælpe dig. Eksemplet herunder viser en 2.4.18 kerne:

```
bruger@debian:~$ uname -r
2.4.18
```

Det kan være en jungle at finde ud af hvilken kerne man skal skifte sin nuværende ud med. Lige så svært kan det være, at finde ud af om man skal compilere en kerne selv, eller om man skal bruge en af de mange prekompilerede kerner, hvor man slipper for alt det "grove" arbejde, men tilgængæld må give afkald på muligheden for selv at bestemme.

I de følgende eksempler bruges en 2.4.23 kerne som eksempel. Men det skulle passe til en hvilken som helst kerne ved blot at udskifte versionsnummeret, med den version du skal bruge.

Du har altså to muligheder når du skal installere en ny kerne. Enten kompilerer du kernen manuelt (se Afsnit 7.9.1, "Kompilering og installation af kerne" [69]), ellers installerer du en prekompileret kerne der følger med Debian (se Afsnit 7.9.2, "Installation af prekompileret kerne" [72]). Den første mulighed giver dig mange flere valgmuligheder når du skal konfigurere kernen, men det er også sværere, og du kan hurtigt komme til at gøre noget galt. Ved den anden mulighed, er installationen nem, men du har kun nogle bestemte kerner at vælge fra, og du har ingen præference for opsætningen af dem.

7.9.1. Kompilering og installation af kerne

7.9.1.1. Nødvendige forberedelser

Før man kan compilere en kerne skal man have kildekoden til en kerne. Denne kan man finde på www.kernel.org [<http://www.kernel.org>] eller et af dets mange mirrors. Kildekoden til en kerne fylder mellem 30-45 MB afhængig af version og typen af den pakkede fil man henter kernen i.

Hvis du allerede har kildekoden til en tidligere kerne kan du med fordel nøjes med at hente en patch til kernen. Læs evt. Afsnit 7.9.1.5, "Patch kernen" [72].

I eksemplet bruger vi en kerne pakket med `.tar.bz2`. Læs eventuelt Afsnit 7.2, "Pakkede filer (tar, zip)" [58] for udpakning af andre formater.

Pak kernen ud i mappen `/usr/src` og lav et symlink til den fra `/usr/src/linux`.

```
debian:~# cd /usr/src
debian:/usr/src# tar jxvf linux-2.4.23.tar.bz2
debian:/usr/src# ln -s linux-2.4.23 linux
debian:/usr/src# cd linux
```

Når man har fået hentet kernen skal man yderligere installere nogle pakker der gør ens Debian system istand til at compilere kerner. Disse pakker installeres med følgende:

```
debian:/usr/src/linux# apt-get install gcc make libc6-dev modutils
```

7.9.1.2. Opsætning

Efter udpakning af kernen, skal den sættes op før vi skal compilere den. Der er en del forskellige grænseflader som kan bruges til opsætningen og nogle behøver ekstra pakker installeret. Metoden som vi benytter vil benytte er **menuconfig**. Herunder kan du se en liste over de forskellige opsætningsmetoder:

config Dette er den mest basale opsætningsmetode. Du skal svare på samtlige spørgsmål som kernen giver dig (der er mange), i en primitiv tekst-grænseflade. Den kan ikke anbefales.

menuconfig Ved at vælge **menuconfig** opsætter du kernen igennem et konsolbaseret menusystem. Dette er meget mere overskueligt end **config** og kan varmt anbefales. Det kræver dog, at du installerer følgende pakke først:

```
debian:/usr/src/linux# apt-get install libncurses5-dev
```

xconfig Denne metode benytter Qt-grænsefladen til at opsætte kernen. Qt kører i X og er derfor en rigtig grafisk grænseflade. Den kræver at nogle ekstra pakker er installeret for at kunne køre:

```
debian:/usr/src/linux# apt-get install libqt3-dev
```

gconfig Denne metode benytter GTK+-grænsefladen til at opsætte kernen. GTK+ kører i X og er derfor, ligesom **xconfig** en rigtig grafisk grænseflade. Den kræver at nogle ekstra pakker er installeret, for at kunne køre:

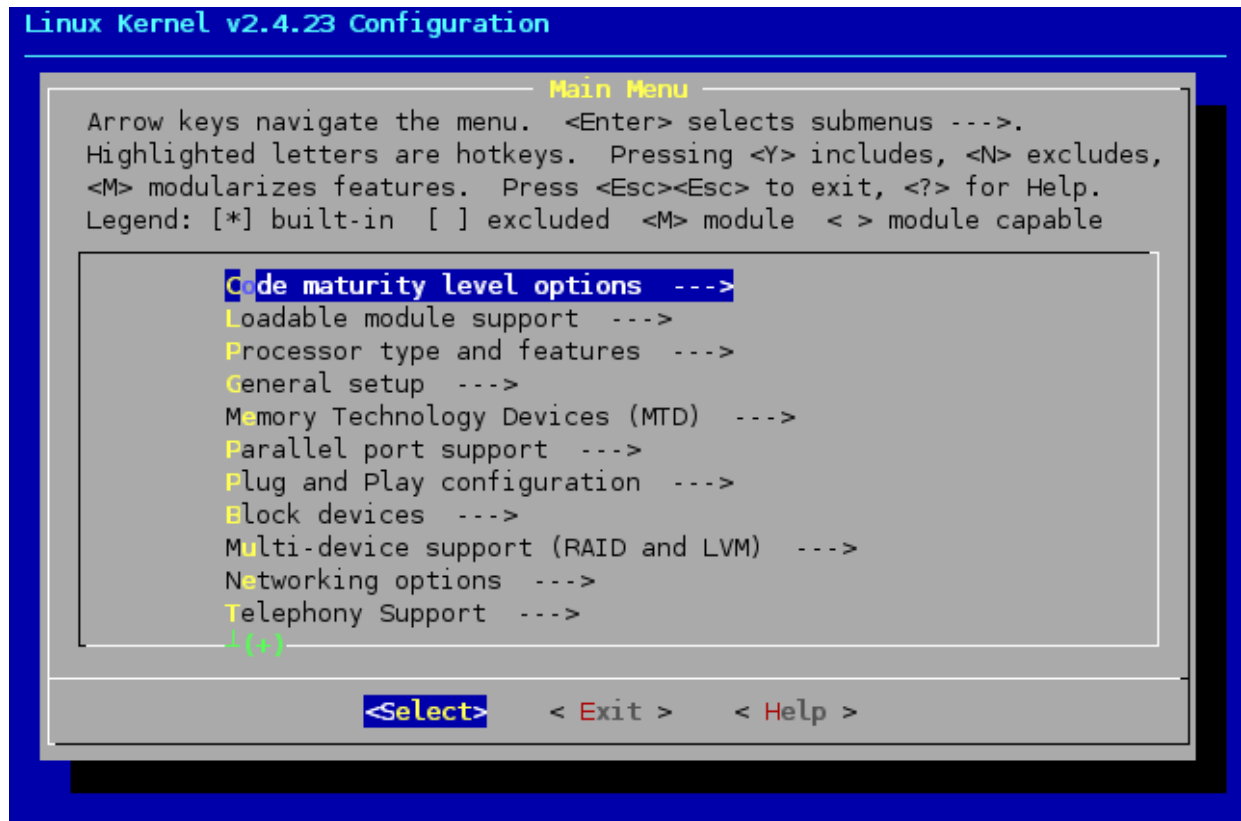
```
debian:/usr/src/linux# apt-get install libgtk2.0-dev
```

For at starte menuopsætning i konsollen skrives:

```
debian:/usr/src/linux# make menuconfig
```

Herefter skulle du gerne se noget tilsvarende den følgende figur.

Figur 7.1. En 2.4.23 kernes menuconfig



Det er vigtigt at du går menuerne igennem flere gange, specielt hvis det er første gang du installerer kernen. Mange af begreberne vil nok forekomme nye, dog er de fleste rimelig godt dokumenteret i menuerne. Yderligere dokumentation kan findes i mappen "Documentation" eller på nettet.

Flere af de mest almindelige problemstillinger er beskrevet i Afsnit 8.3, "Kerne-kompilering og drivere" [80].

7.9.1.3. Kompilering

Til Debian er der blevet udviklet nogle specielle værktøjer til at kompilere kerner med. Værktøjerne, som kan findes i pakken `kernel-package` gør kompileringen af kernen enkel og fremtidig vedligeholdelse af denne let. Hvis man ikke er overbevist og alligevel vil kompilere kernen på traditionel vis, henvises der til `/usr/src/linux/README`.

Det der gør værktøjerne i `kernel-package` så overlegne på et Debian system er at kernen og dertilhørende moduler pakkes ned i en debianpakke. Denne kan så installeres (og fjernes) lige så let som en hvilken som helst anden debianpakke. Den opdaterer endda bootloaderen (LILO), så man efter en genstart automatisk starter op med den nye kerne.

Ved at have sin kerne liggende i en debianpakke er det også nemt at flytte kernen til en anden maskine eller bare gemme kernen så man ikke skal kompilere forfra efter en eventuel geninstallation. Desuden holder man sig til pakkesystemet, og derfor vil du ikke komme ind i den situation at du har filer liggende på dit system som er ude af pakkesystemets kontrol.

Det første du skal gøre, er at installere Debians værktøjer til kernekompileing.

```
debian:/usr/src/linux# apt-get install kernel-package
```

Efter opsætningen af kernen er man blevet bedt om at køre **make dep**, dette ignoreres da den traditionelle kompileringsmetode ikke benyttes. Istedet skrives:

```
debian:/usr/src/linux# make-kpkg --revision=mitimage.1.0 kernel_image
```

Kommandoen **make-kpkg** kompilere kernen og pakker den ned i en debianpakke. Med parameteren **--revision** har man mulighed for at give debianpakken et revisionsnummer (eller et navn) efter eget valg. Ved at udnytte dette er det nemmere at holde styr på forskellige kompileringer af kernen.

7.9.1.4. Installation

I tilfælde af at noget skulle gå galt når maksinen genstartes med den ny kerne, er det vigtigt at du har taget backup af alle dine vigtige data inden du går videre. Skulle du komme ud for at geninstallere hele systemet, mister du ikke noget. Af samme grund er det også en god ide at have en bootdiskette parat, som du burde have lavet under installationen.

Installere debianpakken med kernen på normal vis med:

```
debian:/usr/src/linux# dpkg -i ../kernel-image-2.4.23_mitimage1.0_i386.deb
```

Herefter bliver kernen og tilhørende moduler lagt de rigtige steder. Samtidig får du mulighed for automatisk at opdatere LILOs opsætningen til at pege på den nye kerne. Hvis du hellere selv vil konfigurere LILO eller bruger en alternativ bootloader kan du bare fravælge det.

Bemærk

Det er ikke altid at kernel-package kan finde ud af at opdatere LILO automatisk. Dette er tilfældet hvis du selv har ændret i `lilo.conf`. Som minimum skal din `lilo.conf` indeholde **image=`vmlinuz`** og **image=`vmlinuz.old`** for at kernel-package selv kan opdatere `lilo.conf`.

Kernen kan selvfølgelig fjernes lige så let igen.

```
debian:/usr/src/linux# apt-get remove kernel-image-2.4.23_mitimage1.0_i386
```

Hvis du får brug for at kompilere kernen igen, gøres det præcis som beskrevet tidligere, med den undtagelse af at rester fra den gamle kernekompilering fjernes først med **make-kpkg clean**.

```
debian:/usr/src/linux# make-kpkg clean
debian:/usr/src/linux# make menuconfig
debian:/usr/src/linux# make-kpkg --revision=mitimage.1.1 kernel_image
```

7.9.1.5. Patch kernen

En patch er en fil der indeholder differencen imellem en eller flere filer, og fylder en del mindre end den fulde kildekode. Derfor kan det godt betale sig at hente en patch både for dig og dem der betaler for at hoste filerne.

En patch til kernen hentes på www.kernel.org [<http://www.kernel.org>], samme sted som kildekoden til kernen. Denne fylder som regel nogle få MB og påføres kernen på følgende måde:

```
debian:/usr/src/linux# bunzip2 -c patch-2.4.24.bz2 | patch -p1
```

7.9.2. Installation af prekompileret kerne

Den hurtigste og nemmeste måde at installere en nyere kerne, er at bruge en af de pakker med kernel-images der følger med Debian. Med woody følger der som, tidligere nævnt, en 2.2.20 og en 2.4.18 kerne.

En liste over de kernel-images der er tilgængelig til installation kan findes ved hjælp af **apt-cache**. Her finder vi listen over alle 2.4.18 kerner i woody:

```
debian:~# apt-cache search ^kernel-image | grep 2.4.18
kernel-image-2.4.18-386 - Linux kernel image for version 2.4.18 on 386.
kernel-image-2.4.18-586tsc - Linux kernel image for version 2.4.18 on Pentium-Classic.
kernel-image-2.4.18-686 - Linux kernel image 2.4.18 on PPro/Celeron/PII/PIII/PIV.
kernel-image-2.4.18-686-smp - Linux kernel image 2.4.18 on PPro/Celeron/PII/PIII/PIV SMP
kernel-image-2.4.18-k6 - Linux kernel image for version 2.4.18 on AMD K6/K6-II/K6-III
kernel-image-2.4.18-k7 - Linux kernel image for version 2.4.18 on AMD K7
kernel-image-2.4.18-1-386 - Linux kernel image for version 2.4.18 on 386.
kernel-image-2.4.18-1-586tsc - Linux kernel image for version 2.4.18 on Pentium-Classic.
kernel-image-2.4.18-1-686 - Linux kernel image 2.4.18 on PPro/Celeron/PII/PIII/PIV.
kernel-image-2.4.18-1-686-smp - Linux kernel image 2.4.18 on PPro/Celeron/PII/PIII/PIV SMP
kernel-image-2.4.18-1-k6 - Linux kernel image for version 2.4.18 on AMD K6/K6-II/K6-III
kernel-image-2.4.18-1-k7 - Linux kernel image for version 2.4.18 on AMD K7
kernel-image-2.4.18-bf2.4 - Linux kernel image for version 2.4.18 (bf variant) on 386.
```

Og af det fremgår det selvfølgelig, at man kan installere en kerne ved f.eks. at skrive:

```
debian:~# apt-get install kernel-image-2.4.18-686
```

Ønsker man at installere en 2.6-kerne på sit woody system, kan man tilføje følgende til `/etc/apt/sources.list`:

```
deb http://www.backports.org/debian woody kernel-2.6
```

Hvorefter man kører **apt-get update** efterfulgt af **apt-cache search ^kernel-image** for at se de nye kernel-images man har mulighed for at installere.

7.10. Tidsindstillet kørsel af programmer

Det er meget dejligt at kunne sætte sin box til at udføre forskellige kommandoer på forskellige tidspunkter. Dette gøres vha. cron. Rent praktisk består cron af to dele. Det brugeren oplever, og den del der rent faktisk udfører de valgte programmer: **crontab** og **cron**. Lad os begynde med et lille eksempel og i løbet af det gennemgå de forskellige muligheder der er i cron.

De fleste systemer fjerner indholdet af `/tmp` hver gang du genstarter. Hvis du ikke genstarter særlig tit, kan der samles en hel del filer i din `/tmp`-mappe. Her ville det være smart hvis man f.eks. en gang om dagen checkede om der var nogle filer der ikke var blevet brugt i tre dage, for derefter at fjerne disse. Dette kan nemt automatiseres og køres gennem cron. For at slette en fil har vi programmet **rm** (også omtalt i afsnittet om kommandoer). For at finde de filer vi skal slette benytte vi os af **find** og får lavet os en kommando der ser således ud:

```
find /tmp \! -type d -atime +3 -exec rm -rf {} \;
```

Du skal ikke tænke for meget over hvad denne kommando gør: Det vigtige i dette afsnit er hvordan vi får den kørt på bestemte tidspunkter.

Vi vil altså gerne have kommandoen kørt en gang hver nat kl. 01:00. Derfor skal vi have rettet vores crontab til at køre denne kommando. Først starter vi **crontab** op (vi kører den som root da vi skal have adgang til at slette andre brugeres filer også):

```
debian:~# crontab -e
```

En editor vil nu åbne crontab-filen så du kan rette i den. Som standard er det normalt vi editoren, hvis du vil ændre dette skal du bruge **export EDITOR=navn** hvor **navn** er navnet på din yndlings editor.

Hver linie i filen beskriver hvilken kommando der skal køres på hvilket tidspunkt. Linien har et bestemt format bestående af seks felter:

```
minut time dag måned ugedag kommando
```

Herunder gennemgår vi parametrene én for én:

1. Minuttal (0-59)
2. Timetal (0-23)
3. Dag på måneden (1-31)
4. Måned (1-12 eller navnet på måneden: jan, feb, mar, apr, may osv.)
5. Ugedag (0-6 hvor 0 er søndag. Derudover kan du benytte: mon, tue osv.)
6. Kommando der ønskes udført

Der kan benyttes nogle specielle værdier for tids-parametrene. Hvis du f.eks. ønsker at samtlige muligheder skal benyttes angives dette med en stjerne (*). Derudover kan du separere flere værdier for hver parameter med et komma (ex: **4,7,13,18**). Derudover kan du også angive hele rækker af tal vha. - (ex: **4-9**). Denne kan kombineres med / der gør det muligt at køre en række med et vist antal steps (ex: **2-10/2** vil give samme effekt som **2,4,6,8,10**).

For nu at vende tilbage til vores egentlige formål, skulle vi have lavet en linie der skulle udføres kl. 01.00 hver nat. Givet det der blev gennemgået i forrige afsnit bliver det så:

```
0 1 * * * find /tmp \! -type d -atime +3 -exec rm -rf {} \;
```

Hvis du ønsker videre læsning kan du kigge i følgende manualsider: `crontab(5)`, `crontab(1)` og `cron(8)`.

7.11. Fjernkontrol af Linux

SSH, som står for Secure SHell, er en sikker protokol-variant af telnet, og kan bruges til at logge ind på en maskine med en allerede eksisterende bruger, uden fysisk at sidde foran den. Hvis du bruger telnet og har mulighed for at skifte til SSH, bør du gøre det så hurtigt som muligt!

7.11.1. Brug af SSH

For at du kan etablere en SSH-forbindelse, kræver det at du har sat en SSH-server op på den pågældende maskine. Programmet findes i "ssh"-pakken, hvor sshd, SSH daemonen (serveren), også følger med. Syntaxen for SSH er følgende.

Du kan angive hvilken bruger du ønsker at logge ind med på den pågældende SSH-server:

```
bruger@debian:~$ ssh bruger@debianname
```

Men du kan også lade være med det. I såfald vil programmet bruge den aktuelle bruger som brugernavn:

```
bruger@debian:~$ ssh hostname
```

Herefter vil din maskine prøve at forbinde sig til **hostname** (som er et rigtigt værtsnavn eller en ip-adresse du angiver). Du vil blive spurgt om password hvorefter du logger ind. Det hele foregår krypteret.

BEMÆRK: Det skal nævnes, af sikkerhedsmæssige årsager, at man ikke bør have en SSH-server kørende, hvis man ikke bruger den. Der er en vis sikkerhedsrisiko ved at have en sådan service kørende, som SSH. For eksempel kan den have exploits (huller der kan udnyttes), som gør at en uautoriseret person kan have mulighed for at logge ind på dit system, eller gøre andre utilsigtede ting.

7.11.1.1. Hvordan man logger ind med SSH, uden at skrive password

Hvis man arbejder meget imellem en eller flere maskiner, kan det være irriterende hele tiden at logge ind med sit password via ssh. Man kan dog gøre noget smart, således at man ikke bliver spurgt om password når man logger ind.

Det fungerer således, at hver bruger har en unik privat og public (offentlig) nøgle. Nøglerne passer sammen, således at den offentlige og den private nøgle matcher hinanden. Den private nøgle er hemmelig og må ikke fås af andre, da den derfor kan bruges til at tilgå dine systemer. Den offentlige nøgle kopieres til andre servere. Du kan se den offentlige nøgle som et nøglehul, mens den private er nøglen der kan låse det op.

Det første du skal gøre, er at genere din private og offentlige nøgle. Der findes 3 forskellige krypteringsformer. 'rsa1' er til version 1 af ssh-protokollen mens 'rsa' eller 'dsa' er til version 2 af denne. Her vil vi genere en dsa-nøgle:

```
bruger@debian:~$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_dsa):
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_dsa.
Your public key has been saved in /home/user/.ssh/id_dsa.pub.
The key fingerprint is:
3a:39:f3:29:2d:60:5d:f4:2c:aa:55:a7:2a:c0:80:6c bruger@debian
```

Bemærk at vi intet password skriver her - fordi vi ønsker ikke at blive spurgt om password når vi logger ind på maskinen. Men du må selvfølgelig gerne skrive et password hvis det er det du vil. Din private nøgle findes nu i `.ssh/id_dsa` i din hjemmemappe, mens den offentlige nøgle findes i `~/ .ssh/id_dsa.pub`.

Nu mangler der kun en sidste ting, nemlig at kopiere den offentlige nøgle til en anden maskine. Her kan vi bruge scp kommandoen:

```
bruger@debian:~$ scp .ssh/id_dsa.pub bruger@hostname:
```

Hvor **bruger** er brugeren på den anden maskine, og **hostname** er værtsnavnet eller ip-adressen til maskinen. Husk det sidste kolon! Ellers vil det ikke virke.

Efter at have udført denne kommando, har brugeren din `id_dsa.pub` fil liggende, men den skal lige omdøbes og kopieres til en anden mappe inden det virker, derfor skal du gøre følgende:

```
bruger@debian:~$ ssh bruger@hostname
bruger@hostname's password:
bruger@hostname:~$ mkdir .ssh
bruger@hostname:~$ mv id_dsa.pub .ssh/authorized_keys2
```

Herefter har brugeren på det andet system din offentlige nøgle, og når du næste gang vil forbinde dig til maskinen med det brugernavn, vil nøglerne passe og du kan logge ind uden at angive et password:

```
bruger@debian:~$ ssh bruger@hostname
bruger@hostname:~$
```

Og vupti, du kan nu arbejde lidt mere effektivt.

Det er muligt at have flere offentlige ssh-nøgler liggende hos en bruger, så folk fra flere forskellige steder kan logge ind uden at skulle kende det samme password. Hvis det er tilfældet, skal de offentlige ssh nøgler tilføjes til `~/ .ssh/authorized_keys2`. Det kan gøres således:

```
bruger@debian:~$ cat id_dsa.pub >> .ssh/authorized_keys2
```

>> betyder at der tilføjes til filen fremfor at overskrive.

7.11.2. PuTTY - en SSH/telnet klient til Windows

PuTTY er en gratis SSH/telnet klient til Windows som du kan bruge ligesom SSH-programmet beskrevet i det forrige afsnit.

Hvor du får fat på PuTTY? Det gør du på PuTTY's hjemmeside [<http://www.chiark.greenend.org.uk/~sgtatham/putty/>]. Husk at hente den til den rigtige CPU-arkitektur (hvis du har et Pentium- eller AMD-system, er det "Intel x86").

Når du starter PuTTY møder du et vindue hvor du kan indstille en masse ting. For det meste er det ikke nødvendigt at ændre noget i indstillingerne, men vi vil dog anbefale dig at du lige kigger listen igennem. I det første indstillings-vindue kan du indtaste en IP adresse eller et værtsnavn, og vælge hvilken protokol du vil bruge. Hvis du skal oprette en forbindelse til en SSH-server skal du selvfølgelig vælge "SSH". Så klikker du bare "Open", og vupti, du er forbundet.

7.11.3. Kopiering af filer med scp

scp står for Secure CoPy, og er et program som bruges til at kopiere filer over SSH-protokollen.

Syntaxen for **scp** er:

```
bruger@debian:~$ scp [-P port] [[user@]host1:]file1 [...] [[user@]host2:]file2
```

For eksempel kopierer følgende kommando filen `/home/foo/example.txt` fra computeren "bar" med brugeren "foo", til den nuværende mappe:

```
bruger@debian:~$ scp foo@bar:/home/foo/example.txt .
```

Man kan selvfølgelig også kopiere den anden vej:

```
bruger@debian:~$ scp bar foo@hostname:~
```

(Ovenstående eksempel kopierer "bar", fra den nuværende mappe til hjemmemappen for "foo" fra "hostname" med brugeren "foo".

7.11.4. Brug af sftp

sftp står for Secure FTP og bruges til at oprette en ftp-session over SSH-protokollen.

sftp minder meget om **scp**, der er dog en forskel. **sftp** giver mulighed for at logge ind med ssh, ligesom man logger ind på en ganske almindelig FTP-server.

Syntaxen for **sftp** er følgende:

```
bruger@debian:~$ sftp [user@]host
```

Og hvis du vil bruge en anden port, end den der er standard:

```
bruger@debian:~$ sftp -oPort=24 [user@]host
```

Kapitel 8. Ofte stillede spørgsmål om...

Før du går i gang med at spørge løs i de forskellige forums, hvori du kan få hjælp, er her en liste over de allermest ofte stillede spørgsmål. Ved først at læse dem igennem undgår du at folk i de forums du spørger i, bliver sure fordi de skal svare på de samme spørgsmål gang på gang.

8.1. Apt

8.1.1.

Hvordan installerer jeg en pakke?

```
debian:~# apt-get install foo
```

Ovenstående eksempel vil installere pakken **foo** på dit system.

8.1.2.

Hvordan afinstallerer jeg en pakke?

```
debian:~# apt-get remove foo
```

Ovenstående eksempel vil fjerne pakken **foo** fra dit system.

8.1.3.

Hvordan opdaterer jeg de allerede installerede pakker?

```
debian:~# apt-get update && apt-get upgrade
```

Ovenstående eksempel vil først opdatere det lokale indeks over hvilke pakker der er tilgængelige, for derefter at opdatere de pakker hvor nyere versioner er tilgængelige.

8.1.4.

Hvordan søger jeg efter en pakke?

```
debian:~# apt-cache search php
```

Ovenstående eksempel vil søge efter pakker hvor hele eller dele af navnet svare til **php**. Denne kommando kan med fordel kombineres med **grep**.

8.1.5.

Hvordan fortæller jeg apt hvilket mirror jeg vil benytte?

```
debian:~# apt-setup
```

Ovenstående eksempel vil give dig en dialogboks hvori du kan vælge hvilket medie du vil benytte, og derefter sætte dette op som apt kilde.

8.1.6.

Min `/etc/apt/sources.list` virker ikke! Hvor kan jeg få en ny?

Få **apt-setup** til at lave en for dig. Du kan evt. starte med at slette din nuværende `sources.list` med:

```
debian:~# rm /etc/apt/sources.list
```

8.1.7.

Hvorfor er pakkerne så gamle?

Du kører højst sandsynligt `stable`. Der udkommer ikke nyt software til `stable` - dog kommer der nogle sikkerheds opdateringer ind imellem. Hvis du vil have sprit-nyt software kan det anbefales at skifte til `unstable` eller `testing`. Dette gøres ved, i `/etc/apt/sources.list`, at rette **stable** til enten **testing** eller **unstable**. Derefter kan du køre følgende kommando for at opgradere:

```
debian:~# apt-get update && apt-get dist-upgrade
```

Se også Afsnit 4.2.2, "Opgradering af distribution via apt" [17].

8.2. Netværk

8.2.1.

Hvorhenne angiver jeg mit IP nummer?

Dette finder du i filen `/etc/network/interfaces`. Hvis du er i tvivl om formatet til denne, kan det anbefales at læse **man interfaces**. Kig også i Afsnit 4.1, "Netværksopsætning" [15].

8.2.2.

Hvordan angiver jeg, at jeg vil benytte dynamisk ip (DHCP)?

Her skal du i filen `/etc/network/interfaces` rette **static** til **dhcp**. Husk også at læse **man interfaces**.

8.2.3.

Hvordan aktiverer jeg ipv6 understøttelse?

```
debian:~# modprobe ipv6
```

Denne kommando indsætter `ipv6` i `/etc/modules`.

Men for at få det til at virke, kræver det du har en `radvd` der er fysisk tilkoblet på netværket.

8.3. Kerne-kompilering og drivere

8.3.1. Jeg har problemer med at starte `make menuconfig`.

```
>> Unable to find the Ncurses libraries.  
>>  
>> You must have Ncurses installed in order  
>> to use 'make menuconfig'
```

Får du ovenstående fejl er det fordi du mangler pakken `libncurses5-dev`.

8.3.2. Hvordan får jeg min IDE brænder til at virke når jeg kompilerer ny kerne?

En IDE brænder opfører sig som en SCSI enhed. Derfor skal man under Linux benytte SCSI emulering for at få sin brænder til at virke. Følgende indstillinger slås til:

```
SCSI Support.  
  
[M] SCSI support  
[M] SCSI CD-ROM support  
[*] Enable vendor-specific extensions (for SCSI CDROM)  
[M] SCSI generic support  
[*] Probe all LUNs on each SCSI device  
[*] Verbose SCSI error reporting (kernel size +=12K)
```

Bemærk at SCSI Support skal vælges før man kan vælge SCSI emulering.

```
IDE, ATA and ATAPI Block devices  
  
<*> Include IDE/ATAPI CDROM support  
[M] SCSI emulation support
```

Det er vigtigt at du vælger SCSI emulering som modul. Hvis du kompilerer både IDE cdrom og SCSI emulering som fast bestanddele af kernen, vil den detektere brænderen som en IDE cdrom enhed og brænderen vil derfor ikke kunne brænde medier.

Til sidst skal man lige finde ud af hvilket drev-navn ens brænder har. Drev-navnet er bestemt ud-fra hvor på ide-bussen enheden sidder. hda er primær enhed på første kanal, hdb sekundær enhed osv. Man kan se det helt præcist med kommandoen 'dmesg', hvis man er i tvivl. Når man har fundet ud af dette (eks. hdc) sættes følgende linje ind i lilo for at sikre sig at brænderen bliver startet som en SCSI enhed:

```
append="hdc=ide-scsi"
```

8.3.3. Hvordan får jeg ext3 til at virke? Og hvad er ext3?

Ext3 er en udvidelse af ext2 filsystemet, som er standard under Debian. Ext3 er i modsætning til ext2 et journaliserende filsystem. Dette betyder at hvis maskinen ikke er blevet lukket ordentlig ned, skal man ikke bruge lang tid på at reparere fejl. Desuden er chancen for at der opstår fejl netop ved dette også mindre.

En af fordelene ved at bruge ext3 i stedet for f.eks. reiserfs eller andre journaliserende filsystemer er at man ikke skal formatere sin harddisk for at opgradere. Desuden er ext3 særdeles solidt, da det bygger på det

meget gennemtestede ext2.

At opgradere sin ext2 drev til ext3 er nemt. Først skal man være sikker på man har installeret ext3 i kernen. Ext3 er understøttet fra og med kerne 2.4.15. Man skal huske at slå "Prompt for development and/or incomplete code/drivers" til, da filsystemet stadig er nyt i standard kernen. Skal man opgradere sin kerne er det en god ide at tjekke afsnittet omkring dette.

File systems

```
[*] Ext3 journaling file system support (EXPERIMENTAL)
[ ] JBD (ext3) debugging support
```

Efter man har sikret sig at kernen understøtter ext3 er det tid til at opgradere:

```
debian:~# tune2fs -j /dev/hdXX
```

Her angives drev-navnet i stedet for hdXX, f.eks. hda1 (check med kommandoen **df** hvis du er i tvivl). Genstarter man ofte kan det være en god ide at ændre hvor tit ext3 er tvunget til at checke din disk. Dette gøres med følgende kommando:

```
debian:~# tune2fs -i 0 -c 0 /dev/hdXX
```

Her angiver "i" antallet af dage og "c" antallet af genstart (0 svarer til ingen check). Til sidst skal man opdatere sin `/etc/fstab` til at bruge ext3 i stedet for ext2.

```
/dev/hda3      /          ext3      defaults,errors=remount-ro    0      1
```

8.3.4. Hvilken driver skal jeg bruge til Sound Blaster pci 128?

Creative Ensoniq AudioPCI 97 (ES1371).

8.4. X og window managers

8.4.1. Hvordan angiver jeg hvilken window manager jeg ønsker at benytte?

```
debian:~# update-alternatives --config x-window-manager
```

Når du kører ovenstående kommando vil du få en liste over de tilgængelige window managers, hvorfra du skal være en.

8.4.2. Når jeg starter X får jeg følgende fejl: "No screens found".

Du kan enten vælge at løse dette problem manuelt, eller forsøge at få Debian til at guide dig igennem. Hvis du vil guides kan du prøve at køre følgende kommando:

```
debian:~# dpkg-reconfigure xserver-xfree86
```

Hvis du mere hører til typen der holder af at sætte sig ind i tingene og gerne vil lære hvorledes du får sat din X optimalt op kan det anbefales at læse **man XF86Config-4** hvor du vil få en gennemgang af hvorledes filen `/etc/X11/XF86Config-4`, som er den konfigurations-fil X retter sig efter, skal struktureres. X's logfil, `/var/log/XFree86.0.log` kan også være ret nyttig at kigge i.

8.4.3. Min PS/2 mus gør ikke som jeg vil i X, den hopper forvirret rundt.

Dette kan være fordi at **gpm** er installeret og har taget "kontrollen" over din mus. **gpm** er en daemon der sørger for at din mus virker i console. Til at starte med, kan du prøve at fjerne gpm med **apt-get remove gpm**. Hvis du stadig ønsker at have gpm installeret samtidig med X, bør du omkonfigurere gpm med **gpmconfig**.

8.4.4. Min X starter automatisk ved opstart af maskinen, hvordan undgår jeg det?

Det er programmet **xdm** (X Display Manager) som er sat til at starte ved boot. Du har nu to muligheder.

Hvis du slet ikke vil bruge grafisk login, kan du fjerne xdm simpelt ved at skrive **apt-get remove xdm**. Herefter kan du starte din X manuelt i console ved at skrive **startx**.

Hvis du ikke vil fjerne xdm, men stadig fjerne den fra opstartsprocessen, kan du skrive **update-rc.d -f xdm remove** som fjerner de symbolske lænker der får xdm til at starte. Hvis du vil tilføje dem igen, skal du skrive **update-rc.d xdm defaults**.

Det er ikke altid xdm der styrer det grafiske login, det kan også være gdm eller kdm. Hvis det er tilfældet skal navnene bare udskiftes med det rigtige i forklaringen foroven.

8.4.5. Min Alt GR knap virker ikke (@£\$ m.m.)

Dette skyldes normalt at du har de forkerte indstillinger i `/etc/X11/XF86Config-4`. For de fleste skal **XkbLayout** stå til 'dk' og **XkbModel** skal stå til pc104. Men det kan også være at dette tal skal ændres til f.eks. pc105 eller pc101.

8.4.6. Hvordan opretter jeg punkter til menuen i X?

Først og fremmest, skal du installere pakken **menu** via **apt-get**. Menu filerne skal ligges i `/etc/menu`. Her er et eksempel, af hvorledes en sådan fil skal kunne se ud:

```
?package(menu):needs="x11" section="Apps/Net/Ftp" \  
title="IglooFTP PRO" \  
command="IglooFTP-PRO"
```

?package(menu) Dette betyder at menu-punktet er en del af pakken **menu**. Eftersom de ting man manuelt tilføjer/kompilerer, ikke har en pakke inde på systemet - bliver man nødt til at vælge en pakke som findes på systemet i forvejen, ellers bliver punktet ikke tilføjet til menuen.

:needs="x11" Dette betyder at den kræver X11.

section="Apps/Net/Ftp" Dette betyder at den er i Apps, derefter Net, derefter FTP.

<code>\</code>	Betyder blot at vi har lavet en linie mere, og at fortolkeren af menuen ikke skal stoppe her, men forsætte til næste linje.
<code>title="IglooFTP PRO"</code>	Hvad skal menu punktet hedde, i dette tilfælde vil det være ganske udemærket med IglooFTP PRO.
<code>command="IglooFTP-PRO"</code>	Hvad skal punktet køre når det blive valgt af brugeren?

Når du har oprettet eller redigeret i dine menu punkter, skal du skrive:

```
debian:~# update-menus
```

Derefter skal du evt. genstarte din window manager før det virker.

Yderligere information kan findes i manualsiderne `menufile(5)` og `update-menus(1)`.

8.5. Teknik

8.5.1.

Min lyd virker ikke. Jeg får fejlen: "Failed to open audio device (/dev/dsp): Permission denied". Hvad skal jeg gøre?

Dit lydkort bruger `/dev/dsp` til at overføre lyd med. Som standard har almindelige brugere ikke læse adgang til dette device. Debian har en speciel brugergruppe som har adgang til dette. Den hedder `audio`. Du kan tilføje en bruger til den gruppe med følgende kommando (`foo` er brugernavnet):

```
debian:~# adduser foo audio
```