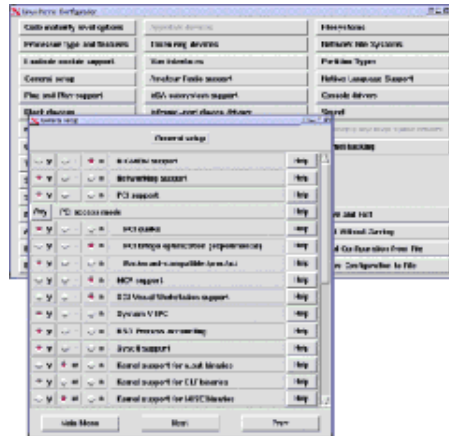




door Philip de Groot
<philipg@authors.linuxfocus.org>

Zelf de Linux kernel compileren



Over de auteur:

Ik promoveer binnenkort aan de Universiteit van Nijmegen op het onderwerp chemometrie. Momenteel werk ik als bioinformaticus op het AMC in Amsterdam en onderhoud daarnaast mijn eigen (Nederlandstalige) Linux beginnerspagina: een van de vele initiatieven voor en door Linux adepten. Ik werk vaak en graag met Linux en schrijf regelmatig mijn ervaringen op.

Kort:

Zo meteen zul je leren hoe je een eigen kernel binnenhaalt, configureert, compileert en installeert.

Inleiding

Waarom wilt U een nieuwe kernel zelf compileren en installeren? Een aantal redenen kunnen zijn:

- De nieuwe kernel heeft betere hardware ondersteuning.
- De nieuwe kernel biedt bepaalde voordelen, zoals betere multiple-processor ondersteuning (SMP-ondersteuning) of de toevoeging van USB ondersteuning in de 2.4.x reeks.
- Programmeerfouten zijn in de nieuwe kernel opgelost.
- Een zelf gecompileerde kernel bevat geen overbodige ballast en is daardoor sneller en stabiel.

Een probleem is dat het zelf compileren van een kernel veel computerkennis vereist. Daarom zal een beginnende Linux gebruiker niet zomaar zelf een kernel proberen te compileren. In dit artikel zijn screendumps gemaakt van de grafische procedure ('make xconfig') die de gebruiker in staat stelt om via een GUI (Graphical User Interface; een grafisch (muis geïntereerd) keuzemenu) de kernel in te stellen. Aan de hand van deze screendumps wordt uitgelegd waarom in specifieke situaties voor bepaalde opties wordt gekozen (of juist niet). In totaal zijn er 40 screendumps gemaakt. Het lijkt wat overdreven om alle

deze screendumps te bespreken, maar het is de beste manier om het functioneren van de kernel en het hoe en waarom van bepaalde kernel opties duidelijk te maken. De screendumps zijn gebaseerd op kernel-2.4.6. Op moment van schrijven is de nieuwste kernel 2.4.19. Op enkele extra ingangen in de menu's na (ondersteuning van nieuwe hardware en dergelijke) zijn de screenshots gelijk en is de kernel compilatie procedure hetzelfde. Een goede raad: print deze pagina voordat je begint, dan heb je altijd alle informatie bij de hand!

De indeling van het artikel is als volgt. Eerst wordt de installatie van de kernel broncode besproken (en waar deze source code op internet gevonden kan worden). Daarna wordt de grafische configuratie van de kernel besproken m.b.v. de screendumps. Als de kernel is geconfigureerd, vindt de uiteindelijke compilatie plaats. De gecompileerde kernel wordt daarna nog niet gebruikt! De kernel moet nog eerst geïnstalleerd worden voor gebruik met de bootmanager 'lilo' of gekopieerd worden naar de partitie waar u 'loadlin' draait (met loadlin kunt u Linux opstarten onder DOS/Windows). Voor gebruik van de bootmanager dient het bestand '/etc/lilo.conf' aangepast te worden en moet het programmaatje 'lilo' gedraaid worden. Dan zijn er nog een aantal specifieke zaken, zoals PCMCIA (specifiek voor laptops). PCMCIA (kleine, creditcard-achtige insteekkaarten met bijvoorbeeld netwerkaansluitingen) is pas sinds kernel 2.4.x in de kernel zelf ondersteund. Voor oudere kernels moet PCMCIA apart gecompileerd en geïnstalleerd worden. Voor SuSE linux is er nog een ander specifiek probleem, namelijk de geluidsondersteuning via de ALSA drivers. Deze drivers zijn niet in de kernel opgenomen en moeten ook opnieuw gecompileerd en geïnstalleerd worden, omdat de originele drivers meestal niet meer werken. Alsof dat nog niet genoeg is kunt U met name bij de overgang naar een nieuwe kernel reeks, bijvoorbeeld van kernel 2.2.x naar 2.4.x, problemen krijgen met een aantal noodzakelijke kernel-utilities, de 'modutils' genaamd. Deze utilities bevatten de programmaatjes om bijvoorbeeld een kernel-module in te kunnen laden (wat modules zijn wordt onder Figuur 3 uitgelegd). Soms kan een nieuwe kernel niet met de oude 'modutils' overweg en dient U een recentere versie te compileren en te installeren. Dit laatste komt echter zelden voor, maar het is wel zo correct om deze 'valkuil' even aan te geven!

Als u nauwgezet de aanwijzingen in dit artikel opvolgt, kan er bijna niets foutgaan. De nieuwe kernel wordt aan 'lilo' toegevoegd of gekopieerd naar de 'loadlin' partitie, waardoor u in geval van nood nog altijd de originele kernel kan opstarten en daarmee kan proberen het probleem op te lossen. Zelfs als u problemen zou ondervinden met de 'modutils' bij een nieuwe kernel kan u nog altijd de originele kernel opstarten en het probleem verhelpen, bijvoorbeeld door de 'modutils' alsnog te compileren en te installeren. Alle nieuwe versies van de 'modutils' zijn 'downwards compatible' met oudere kernels, waardoor ze altijd werken.

Installeren van de kernel broncode

Voor alle handelingen hebt u 'root' privileges nodig, dus u dient als 'root' ingelogd te zijn op het systeem. Eerst en vooral dient u de kernel broncode (source code) te installeren, bijvoorbeeld vanaf de SuSE installatie CD. Deze staat in het onderdeel 'd' ('development') als pakket 'lx_kernel'. Het is aan te raden om de kernel broncode van uw distributie te installeren, omdat dan de grafische hulpmiddelen automatisch meegeïnstalleerd worden. Daarna kunt u een tarball-archief met de nieuwste Linux kernel downloaden (via www.kernel.org/pub/linux/kernel/v2.4/) en installeren, bijvoorbeeld het bestand 'linux-2.4.6.tar.bz2'. De modutils vindt u op www.kernel.org/pub/linux/utils/kernel/modutils/v2.4/. Merk op dat de versie nummers van de 'modutils' niet overeenkomen met de versie nummers van de

kernel. Download gewoon van beide pakketten de meest recente versies. Het compileren en installeren van de 'modutils' wordt onder het kopje 'het installeren van de modutils' behandeld.

De kernel broncode staat doorgaans in de directory '/usr/src/linux/' geïnstalleerd, waarbij de directory 'linux' de huidige kernel source bevat. Het is altijd verstandig om deze kernel broncode te bewaren, bijvoorbeeld door de linux-directory te hernoemen:

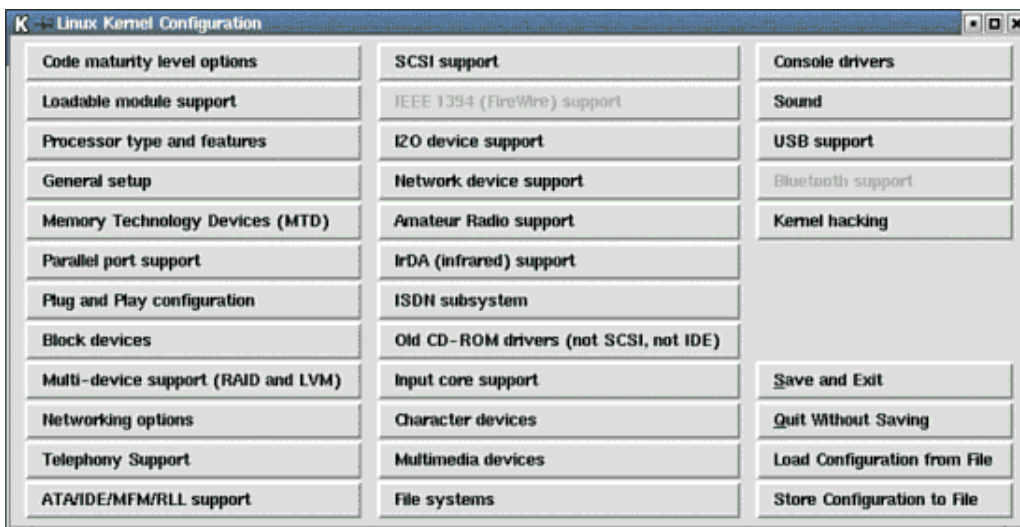
```
cd /usr/src
mv linux linux-2.2.18 (indien uw broncode versie 2.2.18 hebt)
```

Vervolgens pakt u de nieuwste versie van de Linux kernel uit. Merk op dat het ingepakte bestand 'linux-2.4.6.tar.bz2' ALTIJD alles uitpakt in de 'linux' directory. Als deze directory bestaat, wordt hij overschreven met alle nadelen vandien (zoals problemen bij het compileren, oude kernel instellingen die u kwijt bent, enzovoorts). In dit voorbeeld hernoem ik de 'linux' directory (na uitpakken van de kernel broncode) naar 'linux-2.4.6' en maak ik een symbolische link met de naam 'linux' die verwijst naar de directory 'linux-2.4.6'. Het voordeel van deze aanpak is dat in een oogopslag gezien kan worden welke kernel versie aanwezig is op het systeem. Bovendien is het makkelijker om een kernel-upgrade te installeren:

```
cd /usr/src
cp ~/linux-2.4.6.tar.bz2 ( ervan uitgaande dat desbetreffend bestand )
                        ( in uw home-directory ('~/') staat )
bzip2 -d linux-2.4.6.tar.bz2 ( dit kan even duren )
tar -xvf linux-2.4.6.tar
mv linux linux-2.4.6
ln -s /usr/src/linux-2.4.6 /usr/src/linux
```

Vervolgens gaat u naar de directory waar de kernel staat:

```
cd /usr/src/linux
make xconfig ( zie Figuur 1 )
```



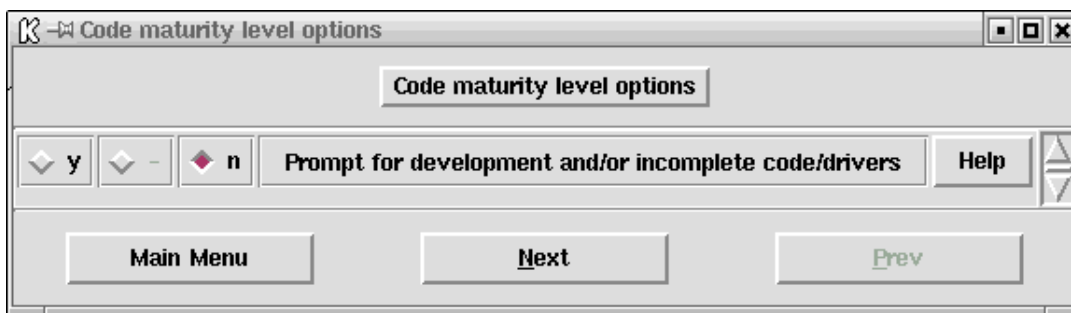
Figuur 1: De grafische interface voor het instellen van uw Linux kernel na de opdracht 'make xconfig'.

Dit is het grafische hoofdmenu waarmee u de kernel kunt instellen. U moet op de verschillende opties klikken om de kernel verder in te stellen. Door te klikken op 'Save and Exit' worden uw instellingen definitief opgeslagen en kunt u de kernel echt compileren en installeren (zie Figuur 40). Zover zijn we nu echter nog niet.

Configureren van de kernel

Hieronder staan de verschillende screendumps met reeds ingevulde selecties (voor mijn persoonlijke situatie). Onder ieder plaatje staat een toelichting over waarom ik bepaalde opties heb gekozen. Door de voorbeelden aandachtig door te lezen begrijpt u waarom bepaalde opties gekozen worden en begrijpt u beter wat u zelf in uw specifieke situatie moet selecteren. U kunt ook een 'make xconfig' draaien onder uw eigen Linux distributie en de (Engelstalige!) help-teksten lezen die tevoorschijn komen als u op 'Help' klikt. In deze helpteksten staat ook vaak of u normaal gesproken de optie aan of uit hebt staan (in geval van twijfel). Deze helpteksten zijn een welkome aanvulling tijdens het configureren van de kernel.

Uiteraard is er bepaalde hardware die u wel heeft, maar die niet in dit voorbeeld behandeld wordt. Desalniettemin wordt het nu wel duidelijk hoe u zelf de hardware kunt instellen (en kan u controleren of bepaalde hardware ondersteund wordt in de kernel door zelf te gaan zoeken).



Figuur 2: Het instellen van de 'code maturity level options'.

In dit onderdeel maakt u de keuze of experimentele kernel opties gebruikt mogen worden of niet. Het kan soms noodzakelijk zijn om deze optie te activeren om hardware ondersteuning te krijgen (van recent verschenen insteekkaarten bijvoorbeeld), maar normaal gezien laat u deze optie uit staan. Experimentele code heeft een negatieve invloed op de stabiliteit van de kernel. In Figuur 1 ziet u de grijze opties 'IEEE 1394 (FireWire) support' en 'Bluetooth support'. Deze twee opties kunt u met de huidige instelling niet selecteren omdat het nog experimentele code betreft.

Figuur 3: Het instellen van de 'loadable module support'.

(vanaf hier worden de screenshots gelinkt, je kan ze openen in een ander venster om ze te bekijken) Modules zijn stukjes kernel code (bijvoorbeeld drivers) die los meegecompileerd worden. Dit betekent dat de code niet in de kernel zelf zit, maar wel ingeladen kan worden (en dus beschikbaar wordt gemaakt aan de kernel) op het moment dat u de code nodig heeft. Algemeen wordt aangeraden om zoveel

mogelijk kernel-code als module te compileren, omdat dan een kleine, stabiele en snelle kernel overblijft. Maak niet de fout om het ext2-bestandssysteem (of ReiserFS als u dat reeds gebruikt) als module te compileren (zie Figuur 32)! De kernel kan dan zijn eigen bestandssysteem niet lezen (dus geen modules of configuratiebestanden laden), hetgeen toch wel een voorwaarde is voor een correct opstarten van Linux! U zult zien dat ik persoonlijk nog relatief weinig met modules werk. Ik ben van mening dat een lopende kernel al de hardware direct moet kunnen aanspreken (zonder modules in te hoeven laden), vandaar. Dit is echter een kwestie van persoonlijke voorkeur.

Figuur 4: Het instellen van de 'processor type and features'.

U selecteert het type processor dat u heeft en geeft aan of de genoemde opties voor u van toepassing zijn of niet. De '/dev/cpu' opties zijn erg geavanceerd en worden door gewone gebruikers niet toegepast. Deze mag u rustig uit zetten. 'High Memory Support' gaat pas spelen als u meer dan 1 gigabyte RAM computergeheugen hebt (NIET de ruimte op de harde schijf) en staat normaal gesproken ook uit. De optie 'Math Emulation' moet u aanzetten als u Linux draait op een 386 of 486SX systeem (dus zonder numerieke coprocessor). Zonder coprocessor kan Linux niet draaien en moet u de emulatie aanzetten. Nagenoeg alle moderne processoren hebben een geïntegreerde numerieke coprocessor, dus deze optie staat standaard uit. Het aanzetten van 'MTRR' is met name nodig als u gebruik maakt van een videokaart die is aangesloten via een PCI of AGP bus. Dit om de snelle datatoevoer beter te laten verlopen. Alle moderne systemen gebruiken een videokaart via de PCI of AGP bus, dus deze optie dient meestal aan te staan. Het is altijd veilig om deze optie aan te zetten, ook als uw systeem geen PCI of AGP bus gebruikt voor de videokaart. Symmetric multi-processing support (SMP) speelt pas als u een moederbord hebt met meer dan 1 hoofdprocessor (bijvoorbeeld een moederbord met 2 pentium II processoren erop). Deze optie zorgt ervoor dat de kernel beide processoren zo optimaal mogelijk belast. De laatste optie (APIC) is ook meestal van toepassing op multiprocessor systemen en staat normaal gesproken uit.

Figuur 5: De algemene kernel instellingen.

Hier specificeert u een aantal algemene instellingen van de kernel. Iedereen zet altijd 'Networking support' aan omdat u dit altijd nodig heeft, bijvoorbeeld voor internet. Linux is een netwerk geïntereerd systeem, dat zonder deze optie niet goed kan draaien. Netwerk ondersteuning is ook voor allerlei andere handelingen nodig die ogenschijnlijk weinig met netwerken te maken hebben. Vandaar dat bij een uitgeschakelde 'Network support' u het risico loopt dat de kernel zich niet laat compileren. Kortom: aan laten staan! Alle moderne systemen gebruiken de PCI bus, dus de PCI-gerelateerde opties zet u ook aan. De grijze tekst 'PCMCIA/CardBus support' kunt u niet selecteren omdat u hebt aangegeven geen experimentele code te willen gebruiken (zie Figuur 2). Als u een laptop gebruikt, kunt u 'PCMCIA/CardBus support' in de kernel gebruiken om bijvoorbeeld een netwerk of een modem insteekkaart operationeel te maken (zie ook het kopje 'pcmcia ondersteuning (laptops)'). 'System V IPC' stelt programma's in staat om te communiceren en te synchroniseren, 'BSD process accounting' bewaart informatie als lopende processen beëindigd worden, zoals de foutcode waarmee een proces stopt en 'Sysctl support' geeft programma's de mogelijkheid om bepaalde kernel-instellingen te wijzigen zonder de kernel opnieuw te compileren of het systeem te herstarten. Uiteraard staan deze opties normaal gesproken aan. De 'kernel core (/proc/kcore/) format' staat voor moderne Linux distributies op 'ELF'. 'ELF' is het formaat waarin verschillende systeembibliotheken zijn geschreven (stukjes programmacode die voor het systeem beschikbaar zijn en door programma's aangeroepen kunnen worden; vergelijkbaar met de windows .dll bestanden). 'ELF' is de opvolger van het verouderde 'a.out' formaat en alle moderne Linux programma's hebben de ELF bibliotheken nodig. Helaas is het voor sommige verouderde programma's onontbeerlijk om de ondersteuning van het 'a.out' formaat mee te nemen. Een voorbeeld hiervan is 'Word Perfect 8 voor XWindows'. Dit is een echte

XWindows/Linux applicatie, maar slechts beschikbaar in het verouderde 'a.out' formaat. Zonder ondersteuning van het 'a.out' formaat, bijvoorbeeld als module, kan 'xwp' eenvoudigweg niet draaien. 'MISC' heb ik ook meegenomen als module. In principe gebruik ik dit niet, maar het is altijd handig om de code beschikbaar te hebben (met name als u vaak 'java' of 'python' gebruik of als u de DOS emulator 'DOSEMU' vaak aanspreekt). 'Power Management support' en 'Advanced Power Management BIOS support' (niet afgebeeld in Figuur 5) heb ik aanstaan om de kernel in staat te stellen de computer automatisch uit te schakelen bij het afsluiten van Linux. Bij de moderne ATX moederborden moet u minimaal deze opties meecompileeren om de computer automatisch uit te kunnen schakelen. De overige power management functies staan uit omdat ze normaal gesproken niet werken onder XWindows (dat bij mij standaard opgestart wordt als ik Linux draai). In KDE en Gnome zijn standaard power management functies aanwezig die u kunt instellen.

Figuur 6: Het instellen van de 'Memory Technology Devices'.

Deze optie hebt u nodig om bijvoorbeeld 'flash kaartjes' uit te kunnen lezen onder Linux. Op dergelijke kaartjes staan bijvoorbeeld foto's die zijn gemaakt met een digitale fotocamera. U kunt dus onder Linux de kaartjes inlezen (als u daarvoor de geschikte hardware hebt) en de foto's als .jpg-bestanden naar de harde schijf kopiëren. Voor een normaal gebruik van de kernel hebt u deze optie niet nodig. In een later stadium kan de gewenste ondersteuning altijd nog meegecompileerd worden.

Figuur 7: Het instellen van de parallele poort.

De parallele poort wordt normaal gezien gebruikt om printers en scanners op aan te sluiten (voordat de USB technologie er was). Ikzelf heb een printer aan de parallele poort hangen en wil dus graag dat deze poort onder Linux beschikbaar is. Let op: door het instellen van de parallele poort hebt u nog niet de mogelijkheid om te printen ingesteld (zie Figuur 28)!

Figuur 8: Het instellen van 'Plug & Play'.

Nagenoeg iedereen heeft een 'Plug & Play' systeem en wilt dus ook ondersteuning hiervoor. Door deze optie aan te zetten kan de kernel de 'Plug & Play' devices zelf configureren en beschikbaar maken aan het besturingssysteem. Soms is het noodzakelijk om in het BIOS de optie 'Plug & Play OS' aan te zetten omdat anders Linux (maar ook Windows) de 'Plug & Play' devices niet zelf kan instellen. De optie 'ISA Plug & Play support' verwijst naar ISA kaarten, zoals de sound blaster AWE64, die wel 'Plug & Play' zijn, maar die gebruik maken van de ISA bus. Voor de ISA bus is nooit een 'Plug & Play' definitie gemaakt, waardoor het automatisch instellen van deze kaarten vaak lastig is. Voordat kernel 2.4.x werd gebruikt moesten Linux gebruikers het programma 'isapnp' aanroepen tijdens het opstarten (uit het pakket isapnptools, doe `rpm -qil isapnptools` om de bestanden hiervan te bekijken). Dit programmaatje las het bestand '/etc/isapnp.conf' in met daarin alle poorten, adressen en interrupts die door de desbetreffende kaarten werden gebruikt. Als 'isapnp' niet werd gedraaid of als het bestand '/etc/isapnp.conf' niet correct was, was de kaart niet beschikbaar voor Linux en werkte bijvoorbeeld het modem, netwerkkaart of de geluidskaart niet. Bovenstaande procedure is nu geautomatiseerd door de optie 'ISA Plug & Play support' aan te zetten (het bestand '/etc/isapnp.conf' wordt niet meer gebruikt; de instellingen worden automatisch correct gedetecteerd). Onder SuSE 7.1 moest ik na het compileren van kernel 2.4.x het bestand '/etc/isapnp.conf' hernoemen (in bijvoorbeeld '/etc/isapnp.conf.old') omdat zowel de kernel als 'isapnp' dezelfde resources claimden met alle gevolgen van dien. Onder SuSE 7.1 (en oudere versies) wordt 'isapnp' namelijk automatisch geactiveerd tijdens het opstartproces (ook al heeft de kernel reeds de benodigde ondersteuning). Gelukkig zijn deze problemen enkel van toepassing op oudere systemen, daar nieuwere distributies standaard de isapnptools niet meer gebruiken.

Figuur 9: Het instellen van 'block devices'.

Nagenoeg iedereen zal een floppy diskette willen gebruiken, dus de bovenste optie staat aan (of in mijn geval als module geselecteerd). Als er een verzoek wordt gegeven tot het benaderen van de floppy, dan laadt de kernel automatisch de benodigde module (als uw distributie het bestand `/etc/modules.conf` of `/etc/conf.modules` goed heeft geconfigureerd, wat normaal gesproken altijd het geval is). U hebt daar als gebruiker geen omkijken naar, op voorwaarde dat u de instellingen in Figuur 3 correct hebt overgenomen. Om de floppy te kunnen benaderen moet de kernel wel het floppy bestandsysteem kunnen lezen. Daarvoor is het belangrijk dat u ook Figuur 32 correct overneemt. De overige opties spelen als u bijvoorbeeld gebruik maakt van IDE opslagmedia die via uw parallelle poort benaderd worden. Deze opties staan normaal gesproken uit, met wellicht de 'loopback device support' als uitzondering. Als u CD's wilt branden onder linux maakt u meestal eerst een image van de CD voordat u de CD daadwerkelijk brandt. Om de inhoud van zo'n image te kunnen bekijken, hebt u de 'loopback device' nodig. Ikzelf heb deze optie (5e regel van onderen) als module geselecteerd (niet weergegeven in Figuur 9).

Figuur 10: Het instellen van 'multiple devices'.

Raid en LVM ondersteuning worden normaal gesproken niet door Linux thuisgebruikers toegepast. Raid houdt in dat het systeem 2 (of meer) harde schijven bevat, waarop de informatie gelijktijdig wordt opgeslagen. Als 1 schijf crasht, draait de andere schijf vrolijk door en het systeem blijft overeind. LVM houdt in dat het mogelijk is om bijvoorbeeld een harde schijf erbij te plaatsen op zodanige wijze dat een bestaande partitie ogenschijnlijk groter wordt. In de praktijk betekent dit dat u niet opnieuw hoeft te partitioneren en een oude (bijvoorbeeld te kleine) partitie niet hoeft te kopiëren naar de nieuwe partitie. Padnamen blijven ook ongewijzigd, enzovoorts. Op zich een handige mogelijkheid, maar niet voor de 'normale thuisgebruiker'.

Figuur 11: Het instellen van 'networking options'.

De optie 'Packet Socket' hebt u nodig om met netwerkdonderdelen te kunnen communiceren zonder dat er een netwerk protocol geïmplementeerd is in de kernel. Over deze optie kan ik kort zijn: altijd selecteren! De meeste opties staan uit, tenzij u bepaalde netwerkdondersteuning nodig heeft. Ik heb bijvoorbeeld 'Network packet filtering (replaces ipchains)' geselecteerd omdat ik de standaard meegeleverde SuSE firewall gebruik. Een firewall beschermt uw computer voor aanvallen van buitenaf, bijvoorbeeld vanaf het internet. Voorwaarde is wel dat u de firewall goed hebt geconfigureerd. Het spreekt voor zich dat het feit dat de beveiliging op kernel-niveau plaatsvindt grote voordelen heeft! De nadere configuratie van 'network packet filtering' staat in Figuur 12 uitgelegd. 'Unix domain sockets' hebt u nodig om netwerk verbindingen tot stand te brengen. Echter: als u deze optie niet activeert, kan XWindows ook niet draaien. XWindows gebruikt namelijk automatisch Unix sockets. Kortom: deze optie moet altijd aan staan! 'TCP/IP networking' bevat de protocollen die op internet en ook vaak op interne netwerken worden gebruikt. Normaal gesproken wilt u TCP/IP ondersteuning geactiveerd hebben. Als u het moeilijk vindt om te beslissen of u een optie wel of niet moet selecteren, kunnen de helpteksten uitkomst bieden. Bij twijfel is het mogelijk om in eerste instantie ondersteuning mee te compileren en in een latere testfase weer te verwijderen. Bepaalde opties als module compileren is natuurlijk ook een goede oplossing.

Figuur 12: Het instellen van het 'IP netfilter'.

Voor SuSE Linux is het voor een correcte werking van de firewall noodzakelijk om de 'terugwaartse' (backwards) ondersteuning voor 'ipchains' in te schakelen. Vandaar. Zie de handleiding van uw distributie voor informatie over uw eigen instellingen (indien u een firewall gebruikt).

Figuur 13: Het instellen van 'telephony support'.

Dit is alleen belangrijk als u een telefonie-insteekkaart in uw computer hebt zitten, bijvoorbeeld om telefoongesprekken over internet te kunnen voeren. Voor de meeste thuisgebruikers staat deze optie uit.

Figuur 14: Het instellen van 'ATA, IDE, MFM en RLL support' (harde schijf communicatie protocollen).

Bijna iedereen heeft deze protocollen nodig om zijn harde schijven aan te kunnen sturen, uitgezonderd systemen die louter SCSI-schijven hebben. Voor de meeste gebruikers zal deze optie geactiveerd zijn. Als u klikt op de regel eronder, krijgt u een submenu met daarin een aantal opties. Deze opties worden hieronder besproken. In totaal zijn er maar liefs 3 screenshots gemaakt. Dit onderdeel is uiterst belangrijk en dient u zorgvuldig in te vullen!

Figuur 15: Het instellen van 'ATA, IDE, MFM en RLL support': screenshot 1.

Iedereen die een apparaat via een IDE/ATAPI interface aanstuurt selecteert de bovenste optie. Dit kunnen zijn: harde schijven, maar ook tape streamers, ZIP diskette stations, CD lezers en CD branders. Alle moderne computers hebben een IDE/ATAPI interface, dus standaard staat de bovenste optie aan. De optie 'include IDE/ATA-2 DISK support' zorgt voor ondersteuning van de harde schijven in een systeem. Deze optie dient dus altijd aan te staan, tenzij u een SCSI-systeem hebt.

Figuur 16: Het instellen van 'ATA, IDE, MFM en RLL support': screenshot 2.

De optie 'include IDE/ATAPI CDROM support' zet u normaal gesproken aan als u een ATAPI CDROM-drive hebt. Als u een CD-brander hebt, is het ook mogelijk om IDE/ATAPI CDROM's te benaderen via SCSI emulatie. Echter, in de praktijk levert dit vaak problemen op, bijvoorbeeld foutmeldingen tijdens het mounten van CD's of tijdens het starten van de CD speler voor het beluisteren van audio CD's. De beste methode is om beide opties, 'include IDE/ATAPI CDROM support' en 'SCSI emulation support' aan te zetten, zoals ook is aangegeven in Figuur 16. Door een aanpassing in '/etc/lilo.conf' kan het apparaat aangewezen worden waarvoor de SCSI-emulatie van toepassing is (meestal de CD-brander). De aanpassing betreft de toevoeging van de regel 'append="hdd=ide-scsi",' zie ook onder het kopje 'Lilo configureren.' Aangezien ik een interne ZIP-drive heb die via een ATAPI interface met mijn moederbord communiceert, heb ik de optie 'include IDE/ATAPI FLOPPY support' aanstaan. Deze optie hebt u ook nodig om andere floppy-achtige drives te activeren, zoals een LS120-drive. De meeste moederborden gebruiken 'PCI IDE' om de harde schijven, cdroms en floppies aan te spreken, dus normaal gesproken staat deze optie aan, evenals de twee mogelijkheden die het gebruik van DMA mogelijk maken. DMA geeft apparaten direct toegang tot het interne geheugen van de computer (dus zonder tussenkomst van de processor), waardoor de IDE schijven sneller te benaderen zijn. Normaal gesproken wilt u dat erg graag! De optie 'sharing PCI IDE interrupts support' staat uit omdat u normaal gesproken deze optie niet nodig heeft. Sommige IDE controllers staan het toe dat hun interrupts gedeeld wordt met een ander computeronderdeel, bijvoorbeeld een exotische netwerkkaart. Aangezien het delen van IDE interrupts prestatieverminderingen van de gekoppelde schijven tot gevolg heeft, wilt u dit normaal gesproken niet (uitgezonderd bij ernstige hardwareproblemen waarbij het delen van interrupts de enige oplossing is).

Figuur 17: Het instellen van 'ATA, IDE, MFM en RLL support': screenshot 3.

Ik heb een Pentium II moederbord met een Intel chipset, dus uiteraard wil ik de specifieke ondersteuning van deze chipset kunnen gebruiken. Als u zelf de kernel opties invult, kunt u kiezen uit andere chipsets die niet in de screenshot (Figuur 17) zijn opgenomen.

Figuur 18: Het instellen van 'SCSI support'.

Als u een SCSI kaart hebt, dient u uiteraard de opties te selecteren die u nodig heeft. In bovenstaande

screenshot staan alleen de opties die u moet selecteren om uw ATAPI CD brander te activeren als u 'SCSI emulation support' hebt geactiveerd (Figuur 16).

Figuur 19: Het instellen van 'I2O device support'.

Als u een I2O interface in uw computer heeft, dient u de benodigde selecties te maken. Normaal gesproken heeft u dit niet en schakelt u 'I2O device support' gewoon uit.

Figuur 20: Het instellen van 'network device support'.

Mijn ervaring is dat de kernel weigert te compileren als 'network device support' uitstaat. Deze optie dient u dus altijd aan te zetten en de dummy driver dient u ook altijd te selecteren (al dan niet als module). De dummy driver wordt vaak gebruikt om Linux goed te laten draaien als er geen fysiek netwerk aanwezig is, zoals bij thuisgebruikers vaak het geval is. Ook als er wel een fysiek netwerk aanwezig is heeft komt de dummy driver nog vaak van pas. In dit grafische menu kunt u ook selecteren wat voor type netwerk en netwerkkaart u heeft. Een voorbeeld wordt in Figuur 21 uitgewerkt. Let op: als u wilt internetten via een modem dient u ppp-support aan te zetten en één van de twee opties 'PPP support for async serial ports' (COM poorten) of 'PPP support for sync tty ports' (voor gehuurde snelle verbindingen over bijvoorbeeld een SyncLink adapter) te selecteren. Doet u dit laatste niet dan krijgt u van de kernel de foutmelding dat de ppp module niet bestaat (terwijl die wel is aangemaakt), wat het erg lastig maakt de echte oorzaak te achterhalen. De beide compressie keuzes kunt u zonder problemen selecteren. Als de kernel ze nodig heeft worden ze gebruikt en anders niet.

Figuur 21: Het instellen van een 'ethernet device'.

In mijn geval is er sprake van een '10 of 100 MBit netwerkkaart' van het merk '3COM'. De netwerkkaart chipset die in mijn situatie gebruikt wordt is de '3c509/3c529'. Aangezien ik geen fysieke verbinding heb met een netwerk (ik heb een netwerkkaart, maar al het netwerkverkeer loopt bij mij over een modem) heb ik de driver als module gecompileerd (voor het geval ik de netwerkkaart ooit eens nodig mocht hebben). Uiteraard selecteert u het type netwerk en de netwerkkaart die u heeft; via uw Linux configuratie programma's ('yast2' onder SuSE) moet u dan nog de netwerkverbinding zelf correct configureren.

Figuur 22: Het instellen van 'amateur radio support'.

Als u gebruik wilt maken van 'amateur radio support' zet u deze optie aan en selecteert u de benodigde driver. Normaal gesproken wordt deze optie niet gebruikt.

Figuur 23: Het instellen van (draadloze) infrarood ondersteuning.

Als u een draadloos apparaat hebt, bijvoorbeeld een draadloze muis, dient u hier de infrarood ondersteuning aan te zetten en de driver te selecteren die u nodig hebt. De meeste desktop systemen hebben geen draadloze muis of toetsenbord en deze optie is dus meestal niet van toepassing.

Figuur 24: Het instellen van de ISDN ondersteuning.

Als u een ISDN-kaart in uw systeem hebt zitten, dient u hier de ondersteuning voor deze kaart te selecteren. Het is belangrijk te weten wat voor soort ISDN-kaart u hebt (chipset etc.) zodat u de juiste driver kan kiezen.

Figuur 25: Het instellen van 'old CDROM drivers'.

In oudere systemen, bijvoorbeeld 486 of zelfs 386 systemen, is de CDROM drive niet via de harde schijf IDE (ATAPI) controller aangesloten, maar via een geluidskaart of een speciale insteekkaart. Om deze oude CD-drives te kunnen gebruiken, dient u de desbetreffende driver te selecteren. Voor moderne

systemen is deze optie achterhaald (en dus overbodig).

Figuur 26: Het instellen van 'input core support'.

Dit is een van de belangrijkste uitbreidingen in de 2.4.x kernels: USB ondersteuning. Het huidige kernel onderdeel is een 'tussenlaag' tussen de kernel en sommige USB apparaten. In Figuur 38 ziet u de mogelijkheid om USB apparaten te selecteren. In de help-teksten van sommige USB apparaten (Figuur 38) staat aangegeven of u de 'input core support' nodig hebt. Indien dit het geval is, dient u de huidige optie aan te zetten. Zoals gezegd is dit onderdeel een 'tussenlaag' tussen de kernel en de uiteindelijke USB ondersteuning. Vandaar de (op het eerste gezicht) wat vreemd aandoende werkwijze. In mijn persoonlijke situatie heb ik 'input core support' niet nodig, dus staat deze optie bij mij uitgeschakeld. Alle moderne moederborden hebben een USB mogelijkheid, dus normaal gesproken zet u deze optie, afhankelijk van uw USB apparaten, aan!

Figuur 27: Het instellen van de 'character devices': screenshot 1'.

De bovenste optie (virtual terminal) geeft u de mogelijkheid om een xterm te openen (XWindows) of in tekstmodus in te kunnen loggen. Deze optie staat normaal gesproken altijd aan! De 2e optie ('support for console on virtual terminal') definieert waarheen kernel boodschappen (bijvoorbeeld meldingen over ontbrekende modules, modules die niet correct functioneren, kernelp Problemen, enzovoorts) gestuurd moeten worden. Onder XWindows hebt u vaak een apart window met de kernelboodschappen, maar in tekstmodus worden deze boodschappen vaak naar de eerste virtuele terminal ('CTRL+ALT+F1') gestuurd, zoals bijvoorbeeld de kernel berichten tijdens het opstarten! Normaal gesproken staat deze optie dus aan. U kunt er ook voor kiezen om deze boodschappen over de seriële poort te verzenden (naar bijvoorbeeld een printer of een terminal; de 4e optie). U dient dan wel de seriële poort te activeren via de 3e optie. Dit moet u ook doen als u de seriële poort wilt kunnen gebruiken, bijvoorbeeld omdat u een seriële muis hebt! De 3e optie ('standard/generic (8250/16550 and compatible UARTs) serial support') staat normaal gesproken aan. Persoonlijk heb ik gekozen voor een compilatie als module. In het SuSE opstartscript wordt geklaagd over het ontbreken van de 'serial support' module tijdens het opstarten, wat ik op deze wijze elegant heb opgelost (door er voor te zorgen dat de module bestaat). Het instellen van de 'character devices' is uiterst belangrijk! Als u dit niet goed doet, kan u met een niet werkend systeem komen te zitten. Vandaar dat in de Figures 28, 29 en 30 nog een aantal overige opties worden besproken.

Figuur 28: Het instellen van de 'character devices': screenshot 2.

Als u op uw eigen machine op afstand een xterm wilt kunnen draaien (bijvoorbeeld via 'telnet' of 'ssh') dan moet u de optie 'unix98 PTY support' aanzetten. Op een standalone desktop systeem hebt u deze optie normaal gesproken niet nodig, maar op de achtergrond lopen toch een aantal processen die deze optie gebruiken. Het is verstandiger om deze optie toch te activeren, al was het maar om foutmeldingen tijdens het opstarten (SuSE) te voorkomen. 'Parallel printer support' heeft iedereen nodig die zijn printer via de parallelle poort heeft aangesloten. Heel vaak is dit het geval, met uitzondering van moderne USB printers. 'Support for console on line printer' creëert de mogelijkheid om de kernel 'console' boodschappen via de parallelle printer te laten lopen. Normaal gesproken wilt u dit niet. De optie 'support for user-space parallel port device drivers' hebt u nodig voor bepaalde apparaten die aan uw parallelle poort zijn aangesloten. Normaal gesproken hebt u deze optie uitstaan. 'I2C support' hebt u meestal niet nodig. Voor sommige video-bewerkingskaarten hebt u deze driver nodig, maar dat kan in een later stadium altijd nog in de kernel geïntegreerd worden (als u een correct werkende kernel hebt). Merk op: I2C is wel nodig wanneer u systeeminfo zoals de temperatuur van de processor en de rotatiesnelheid van de ventilatoren wilt kunnen lezen. In dat geval moet u uiteraard ook 'hardware sensors support' activeren. De opties voor muis- en joystickondersteuning selecteert u wanneer dit voor

u van toepassing is. Niet alle muizen hebben deze muisdriver nodig (dit is in Figuur 29 verder besproken). 'QIC-02 Tape support' slaat op de ondersteuning van dit type tape-drives. Met de huidige CD branders worden nog weinig tape-drives gebruikt, waardoor deze optie meestal uitstaat.

Figuur 29: Het instellen van de 'character devices': Mice.

Als u een seriële muis gebruikt, hoeft u in dit onderdeel niets in te stellen. Voor alle andere muistypen moet u hier wel de benodigde parameters instellen. Als u gebruik maakt van een ORIGINELE bus muis, dan selecteert u de bovenste optie met (daaronder) het bijbehorende type (of merk) busmuis. Tegenwoordig hebben veel computers een ander type muis, die ten onrechte 'busmuis' of 'PS/2 muis' worden genoemd. Dit zijn muizen die vaak via '/dev/aux' zijn aangesloten via een klein plugje, gelijk aan de toetsenbordplug. Deze muis is ook vaak naast de toetsenbordplug (op de computer) aangesloten. Om deze muis correct te laten functioneren, selecteert u de opties 'mouse support (not serial and bus mice)' en 'PS/2 mouse (aka "auxiliary device" support)', zoals in Figuur 29 is voorgedaan.

Figuur 30: Het instellen van de 'character devices': screenshot 3.

De opties die in de kernel configuratie tussen Figuur 28 en Figuur 30 staan en die hier niet worden behandeld, zet u normaal gesproken uit. De optie 'Ftape, the floppy tape device driver' slaat op ondersteuning voor tape drives die via de floppy controller zijn aangesloten. Ook al hebt u zo'n tapedrive, dan nog is het niet essentieel om deze ondersteuning (in eerste instantie) mee te compileren. De overige opties slaan op moderne 3D videokaarten. Als u een videokaart hebt die via een AGP-bus is aangesloten, kan u de AGP-ondersteuning aanzetten en de specifieke driver voor uw videokaart (onder '/dev/agpgart (AGP support)') aanzetten. Merk op dat het mogelijk is om een probleemloos werkend linux systeem te hebben zonder deze opties, maar niet noodzakelijkerwijs. Mensen die een moederbord hebben met een geïntegreerde videokaart (zoals de intel i815 chipset) MOETEN de kernel driver gebruiken! Zo niet, dan zal XWindows versie 4.0 of hoger (gebruikt bij de meest recente distributies) niet werken. In mijn persoonlijke situatie heb ik wel een AGP-kaart, maar deze kaart (een NVidia TNT2 kaart) wordt niet door een specifieke kernelmodule ondersteund (NVidia weigert hardware specificaties openbaar te maken, noodzakelijk voor de ontwikkeling van drivers). Voor mij heeft het dus (helaas) weinig zin om de AGP ondersteuning aan te zetten. Gelukkig werkt bij mij XFree 86 4.0 wel zonder de kernel driver. 'Direct rendering support' slaat op een nieuwe optie vanaf XWindows 4.0 voor een versnelling van de grafische prestaties via de kernel. Ook hier geldt de voorwaarde dat uw specifieke videokaart ondersteund dient te zijn (en dat u XFree86 4.0 of hoger draait). Bovendien moet 'AGP support' ook geactiveerd zijn. Voor een correct functioneren van de Linux kernel mag u ook deze opties rustig uit laten staan.

Figuur 31: Het instellen van de 'multimedia devices'.

Deze optie zet u aan als u een video bewerkingskaart of een radiokaart in uw computer hebt zitten. Uiteraard is dit onderdeel niet essentieel voor een correct functioneren van de kernel.

Figuur 32: Het instellen van de 'file systems': screenshot 1.

In dit onderdeel selecteert u welke bestandssystemen de Linux kernel moet kunnen lezen. Dit is belangrijk om bijvoorbeeld Windows schijven (of diskettes) te kunnen lezen, maar ook het eigen Linux ext2 bestandstelsysteem (of het nieuwe ReiserFS) moet u hier selecteren. Als u dat niet doet, kan Linux eenvoudigweg niet opstarten (omdat de kernel niet vanaf zijn eigen opstartschijf kan lezen). Dit is ook al in Figuur 3 aan de orde geweest. De optie 'DOS FAT support' hebt u nodig om DOS/Windows diskettes en schijven te kunnen lezen (dit geldt overigens niet voor Windows NT/Windows 2000 schijven; hiervoor is een aparte (alleen lezen) driver aanwezig wat verderop in deze opties). De optie 'MSDOS fs support' hebt u nodig om DOS/Windows schijven en diskettes te kunnen lezen en schrijven. Nagenoeg

iedereen wilt dit, dus deze opties staan meestal aan. 'VFAT' is de ondersteuning voor het lezen van en het schrijven naar de lange bestandsnamen onder Windows-95 of 98. Mijn persoonlijke systeem is een zogenaamd 'dual boot' systeem, wat betekent dat ik zowel windows 98 als Linux kan opstarten (via de Linux partitiemanager lilo, zie het kopje 'Lilo configureren'). Vandaar dat ik ook deze optie heb geactiveerd. Als u CD's wilt kunnen lezen die in het gangbare (windows) formaat zijn geschreven, wat officieel de ISO 9660 standaard wordt genoemd, dient u deze optie te activeren. De optie daaronder, de Joliet extensies, zijn een toevoeging van Microsoft aan de ISO 9660 standaard en maakt het mogelijk om ook lange bestandsnamen op CD's te gebruiken. Volgens de ISO 9660 standaard kunnen alleen de officiële MS-DOS 8.3 bestandsnamen gebruikt worden. Vrijwel iedereen wil de hedendaagse CD's kunnen lezen, dus deze beide opties staan normaliter aan. In Figuur 33 worden nog wat overige opties, waaronder het Linux bestandssysteem ext2, belicht.

Figuur 33: Het instellen van de 'file systems': screenshot 2.

Het '/proc' bestandssysteem geeft u informatie over de status van het systeem, bijvoorbeeld over welke interrupts in gebruik zijn. Deze optie staat normaal gesproken altijd aan. Het 'Second extended fs support' is het standaard Linux bestandssysteem zoals dat nu (nog) wordt gebruikt. Dit MOET u zondermeer meecompileeren (NIET als module)! 'ReiserFS' (de optie is niet zichtbaar in de Figuren 32 en 33, maar kan wel in dit onderdeel geselecteerd worden) is een alternatief voor het 'ext2' bestandssysteem en is beter bestand tegen beschadigingen van de bestandsstructuur door bijvoorbeeld het plotseling uitvallen van de computer en dergelijke. Momenteel is ReiserFS nog in ontwikkeling en is daarom ook als experimentele code aangemerkt. Desondanks ondersteunen de meeste recente distributies het gebruik van ReiserFS en in de toekomst zal dit bestandssysteem 'ext2' vaak gaan vervangen. Voor dit moment raad ik persoonlijk het gebruik van ReiserFS nog even af. 'UDF file system support' hebt u nodig als u onder windows het programma packetCD gebruikt, waarmee u 'on the fly' bestanden kan kopiëren (en lezen) van CD's. Dit maakt van een CD als het ware als een trage harde schijf, erg handig om informatie uit te kunnen wisselen met andere PC's. Om deze packet CD's ook onder Linux te kunnen lezen, moet u deze CD's mounten met het 'udf' bestandssysteem (bijvoorbeeld met de opdracht 'mount -t udf /dev/scd0 /cdrom'). In dit onderdeel moet u ook nog zaken regelen als 'Network file systems', 'partition types' en 'Native language support'. Met 'Network File Systems' hebt u normaal gesproken niet te maken, tenzij uw computer onderdeel uitmaakt van een groot netwerk. Dan dient u 'NFS File System Support' en wellicht ook nog 'SMB file support' (voor Windows-netwerken) te activeren. Om Linux op een computer te kunnen draaien, hebt u deze opties niet nodig. De optie 'Partition Types' is nogal geavanceerd en is niet noodzakelijk om de Linux kernel goed te kunnen gebruiken. Deze optie kunt u het beste uitschakelen. 'Native Language Support' wordt in de onderstaande 2 figuren (Figuur 34 en Figuur 35) verder uitgelegd.

Figuur 34: Het instellen van 'native language support': screenshot 1.

In dit menu selecteert u welke codetabel Linux moet gebruiken om DOS/Windows bestandsnamen goed te kunnen afbeelden. De codetabellen die in Figuur 34 staan afgebeeld hebben betrekking op de DOS bestandsnamen. Indien u gebruik wilt kunnen maken van de langere bestandsnamen, dan dient u de NLS-tabellen van Figuur 35 te gebruiken. De bovenste optie in Figuur 34, 'Default NLS option', bepaalt welke tekens Linux standaard moet kennen. De optie 'iso8859-15' staat afgebeeld (en uitgelegd) in Figuur 35.

Figuur 35: Het instellen van 'native language support': screenshot 2.

De 'NLS ISO 8859-15' optie hebt u nodig om het Windows FAT en (CDROM Joliet) bestandssysteem correct weer te kunnen geven (dus de juiste karakters in de bestandsnamen te verkrijgen; altijd handig natuurlijk). De selectie 'NLS ISO 8859-15' vertegenwoordigt Westerse talen inclusief de Euro en deze

codetabel moet nagenoeg altijd meegecompileerd worden. Tabel 'NLS ISO 8859-1' is de oude tabel voor de Westerse talen zonder het Euro-symbool.

Figuur 36: Het instellen van de 'console drivers'.

'VGA text console' geeft u de mogelijkheid om de tekstmodus te gebruiken in VGA resolutie. Nagenoeg iedereen wilt deze mogelijkheid, dus deze optie staat altijd aan. Alle moderne computers (uitgezonderd misschien wat oude 386 computers) hebben een VGA (compatible) kaart en hebben geen enkel probleem met deze instelling. De tweede optie, 'video mode selection support', geeft u de mogelijkheid om tijdens het opstarten van de kernel op te geven in welke resolutie de Linux tekstmodus moet draaien. Dit kan handig zijn als u meer karakters op een regel kwijt wil, maar normaal gesproken laat u deze optie uit. De overige twee opties zijn experimenteel en het gebruik daarvan raad ik af.

Figuur 37: Het instellen van het geluid.

In dit onderdeel dient u de geluidsondersteuning in te stellen. Als uw distributie de ALSA geluidsdrivers gebruikt (zoals SuSE vanaf versie 6.3 en hoger), dan hoeft u alleen maar 'sound card support' als MODULE te selecteren. De ALSA drivers worden later gecompileerd en kunnen dan met de Linux kernel samenwerken (zie ook het kopje 'SuSE en de ALSA geluidsdrivers'). Als uw distributie de standaard kernel geluidsdrivers gebruikt, dient u hier de juiste driver(s) voor uw geluidskaart te selecteren. Nagenoeg alle merken geluidskaarten staan hier genoemd, dus het selecteren van de juiste driver zou geen probleem mogen zijn. Als uw geluidskaart goed werkt met de standaard kernel die bij uw distributie zit, kan u ook het configuratieprogramma (bijvoorbeeld 'yast2' voor SuSE) gebruiken om te achterhalen welke drivers voor uw geluidskaart worden gebruikt. Al met al is dit onderdeel niet kritiek voor een correcte werking van de kernel. Als hier iets mis gaat, hebt u geen geluid (maar dat is dan ook alles).

Figuur 38: Het instellen van 'USB support'.

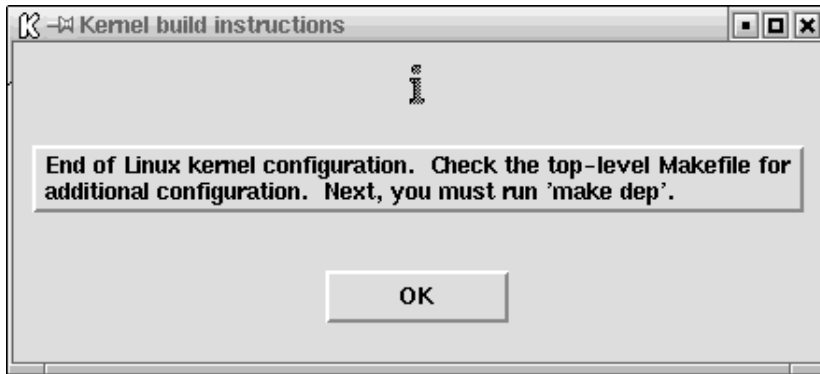
In mijn persoonlijke situatie heb ik wel een USB mogelijkheid op mijn moederbord, maar maak ik er geen gebruik van. Echter, als ik alle USB support zou uitschakelen, dan krijg ik een foutmelding van SuSE tijdens het opstarten. SuSE biedt uiteraard standaard USB ondersteuning en probeert dus logischerwijs de benodigde module(s) te laden (vandaar dat ik ook 'Support for USB' als module heb geselecteerd). Om die (voor mij onbelangrijke) foutmelding elegant op te lossen, compileer ik de driver die nodig is om mijn USB-poorten op het moederbord te kunnen benaderen. Minimaal dient u daarvoor 'Preliminary USB device filesystem' op 'y' te zetten en een specifieke USB-driver te laden. Aangezien ik een wat verouderd Pentium II moederbord heb, heb ik de 'UHCI (Intel PIIX4, VIA, ...)' driver als module geselecteerd. Als u een recent moederbord hebt met een Intel chipset, dan dient u de 'UHCI Alternate Driver (JE) support' module te selecteren. Voor o.a. Compaq computers dient u 'OHCI support' als module te selecteren. In principe hebt u slechts 1 van deze 3 modules nodig, maar in geval van twijfel mag u ze alledrie selecteren. Uw Linux distributie heeft tijdens de installatie al achterhaald welke driver u nodig hebt en laadt automatisch de correcte module.

Met alleen het beschikbaar maken van de USB-poorten op uw moederbord bent u er nog niet. U dient namelijk ook drivers (modules) op te geven van de USB-apparaten die u op uw computer hebt aangesloten (uit de lijst die onder 'USB Device Class drivers' staat). Dit is een rechttoe-rechtaan onderdeel waar weinig mis kan gaan. Lees in geval van twijfel altijd de help-teksten.

Figuur 39: Het instellen van 'hacking'.

Over deze optie kan ik kort zijn: NIET selecteren! Dit wordt door programmeurs gebruikt om bij een kernel crash nog bepaalde zaken uit te proberen (zoals de cache van de harde schijf leegmaken) of om de

oorzaak van een kernel crash te achterhalen. Voor normale gebruikers heeft deze optie geen enkel nut.



Figuur 40: Save and Exit.

Oef! We zijn er doorheen geploegd. Nu de kernel nog even compileren en installeren. Dit wordt hieronder verder behandeld.

De kernel compileren

Als u op 'Save and Exit' hebt geklikt, worden de instellingen opgeslagen in het bestand './config' ('/usr/src/linux/.config'). Even terzijde: als u een kleine kernel upgrade doet, bijvoorbeeld van kernel 2.4.5 naar kernel 2.4.6, dan is het handig om dit bestand even te bewaren en naar de nieuwe kernel source te kopiëren. Al uw oude instellingen zijn (meestal) bewaard, hetgeen u veel werk kan besparen. Op dezelfde manier is het soms handig om te starten met de configuratie van de 'standaard' kernel die in je distributie zit, waarvan je het bestand vaak kunt terugvinden als /boot/config (kopieer het naar ./config om het te gebruiken). Maar let op, als u de kernel broncode hebt vernieuwd, en u om onduidelijke redenen problemen krijgt met het compileren van de kernel, verwijdert u dit bestand (uiteeraard) als eerste! Natuurlijk hebt u veiligheidshalve de instellingen van uw correct werkende kernel opgeschreven en zorgvuldig bewaard. De procedure om vervolgens de kernel te compileren is als volgt:

```
make dep
make clean ( voor oudere kernels )
make bzImage
make modules
make modules_install
```

Dat u de opdracht 'make dep' moest uitvoeren was in Figuur 40 al aangegeven. Uiteraard voert u deze commando's uit in de Linux source directory, normaal gesproken: '/usr/src/linux'. Voor oudere kernels uit de 2.0.x reeks (of ouder) moest ook nog de opdracht 'make clean' opgegeven worden, zodat verouderde bestanden verwijderd worden voordat een nieuwe kernel wordt gecompileerd. Deze opdracht was noodzakelijk om rare, niet te achterhalen, foutmeldingen te voorkomen (omdat verouderde object (.o) bestanden niet altijd werden vernieuwd). Met de opdracht 'make bzImage' wordt een nieuwe kernel gecompileerd maar nog niet geïnstalleerd. Er bestaan andere 'make' opdrachten om een kernel te compileren, zoals 'make bzImage' of 'make zImage,' maar met deze opdrachten kunt u op onverwachte problemen stuiten. De meeste kernels zijn te groot om een 'make zImage' goed te laten verlopen: tijdens

compilatie komt er een foutmelding en hebt u geen kernel. Bij de opdracht 'make bzlilo' dient het een en ander goed ingesteld te zijn in bijvoorbeeld '/etc/lilo.conf,' wat niet altijd het geval is. Vandaar dat deze twee opdrachten veiligheidshalve niet worden gebruikt. Met de opdracht 'make modules' worden de modules gecompileerd maar nog niet geïnstalleerd. Daar zorgt de opdracht 'make modules_install' voor die de modules in de directory '/lib/modules/2.4.6/' zet. 2.4.6 staat voor de huidige kernelversie en verandert als u een andere kernelversie compileert. Op deze wijze worden de bijbehorende modules automatisch in een aparte directory gezet zodat u nooit conflicten kunt krijgen met verouderde modules en dergelijke. Tijdens het opstarten 'weet' de Linux kernel in welke directory de gewenste modules staan. Als u kernel 2.4.6 al eerder hebt gecompileerd en nu opnieuw hercompileert worden de bestanden wel overschreven en blijven eventuele oudere modules staan (als ze volgens de kernel configuratie niet meer nodig zijn)! Normaal gesproken is dit echter geen probleem, maar het is altijd verstandig eerst even de oude modules te verwijderen voordat u de nieuwe installeert.

Om de kernel veilig te kunnen installeren, dient het lilo-configuratiebestand '/etc/lilo.conf' (eventueel) aangepast te worden en moet u de kernel en het bestand 'System.map' naar de correcte locatie kopiëren. Alsof dat nog niet genoeg is, moet u ook nog de opdracht 'lilo' uitvoeren. De alternatieve werkwijze is het gebruik van 'loadlin', waarmee u een Linux kernel onder Dos/Windows kunt opstarten. Beide opties worden hieronder verder uitgewerkt.

Lilo configureren

Het 'lilo' configuratiebestand vindt u meestal in de '/etc' directory als '/etc/lilo.conf'. Dit bestand kunt u met een normale ASCII editor openen, bijvoorbeeld met 'xemacs /etc/lilo.conf &' (als u 'XEmacs' hebt geïnstalleerd, wat tijdens de meeste installaties niet gebeurt). Vaak kunt u ook 'emacs' gebruiken of eventueel een andere teksteditor. Een 'lilo' configuratiebestand ziet er bijvoorbeeld als volgt uit:

```
boot      = /dev/hda
vga       = normal
read-only
menu-scheme = Wg:kw:Wg:Wg
lba32
prompt
timeout = 300
message = /boot/message

    other = /dev/hda1
    label = win98

    image = /boot/bzImage
    label = linux-2.4.6
    root = /dev/hda3
    append = "parport=0x378,7 hdd=ide-scsi"

    image = /boot/vmlinuz.suse
    label = suse
    root = /dev/hda3
    append = "hdd=ide-scsi"
    initrd = /boot/initrd.suse
```

Het bovenstaande kan op ieder systeem (en iedere Linux distributie) afwijken. Vandaar dat ik nu eerst door dit configuratiebestand heenwandel. De bovenste 8 regels staan al goed en hoeft u normaal gesproken niet te veranderen. De 'boot' opdracht in de eerste regel geeft aan vanaf welke fysieke hardeschijf opgestart wordt (feitelijk verwijst 'boot' naar de locatie van de 'master boot record'). In mijn situatie dus vanaf /dev/hda: de eerste fysieke harde schijf. De optie 'vga' geeft aan dat een standaard VGA tekst modus wordt gebruikt met 80x25 karakters. De optie 'read-only' betekent dat de Linux partitie bij het opstarten eerst read-only wordt gemount. Tijdens het opstarten van Linux worden de partities gecontroleerd op fouten. Pas daarna worden de partities met zowel lees- als schrijfrechten opnieuw gemount. 'menu-scheme' regelt de kleuren van het 'lilo' menu tijdens het opstarten in de tekstmodus. Met 'lba32' is er een mogelijkheid om een besturingssysteem na de 1024ste cilinder (van de harde schijf) te booten op voorwaarde dat het BIOS dit ondersteunt (alle moderne systemen ondersteunen 'lba32'; zoniet kan een BIOS upgrade een oplossing bieden). Met de grote harde schijven van tegenwoordig is dit beslist geen overbodige luxe! 'prompt' dwingt 'lilo' met een prompt te komen zodat de gebruiker een besturingssysteem naar keuze kan selecteren. De optie 'timeout' geeft aan hoeveel milliseconden 'lilo' wacht op invoer aan de prompt voordat het standaard besturingssysteem wordt opgestart. Als er geen standaard besturingssysteem is ingesteld (zoals in bovenstaand voorbeeld) wordt het eerste besturingssysteem opgestart. In mijn geval is dat Windows98, zodat mensen die niet goed met Linux overweg kunnen standaard in een Windows omgeving terecht komen. De 'message' optie laat een bericht zien tijdens het uitvoeren van 'lilo'. Onder SuSE is dit een leuke Linux Pinguin als achtergrond met (uiteraard) de tekst 'SuSE Linux 7.1' (typt u maar eens in 'xv /boot/message' of 'gv /boot/message', of soms zelfs 'gimp /boot/message'; 'xv' of 'gv' (GhostView) zijn een shareware programma's waarmee u ondermeer allerlei afbeeldingen kunt bekijken). Merk op dat het bestand /boot/message niet aanwezig is op systemen die geen grafisch login-scherm hebben (b.v. oudere distributies), in dat geval zal de 'boot message' gewoon een stuk tekst zijn. In principe kunt u trouwens zelf een plaatje laten afbeelden tijdens het opstarten via 'lilo' (heb ik niet uitgetprobeerd)! Alle 'lilo' opties kunt u met de opdrachten 'man lilo' en 'man lilo.conf' nalezen in de manual pages van uw Linux systeem.

De overige opties regelen het opstarten van de verschillende besturingssystemen. U kunt maximaal 16 verschillende besturingssystemen of kernels laten opstarten, wat normaal gesproken ruim voldoende is. Met de regel 'label=' wordt de naam van het besturingssysteem geselecteerd. Windows98 (en ook oudere Windows versies en DOS, maar niet Windows NT of Windows 2000) staan altijd op de eerste primaire partitie, dus alleen een 'other' regel en een 'label' regel zijn voldoende voor een correct opstarten van deze besturingssystemen. In de tweede sectie (vanaf 'image=/boot/bzImage') wordt de nieuwe kernel ingesteld met als label 'linux-2.4.6'. Mijn Linux root directory is '/dev/hda3'. Met de regel 'append = "parport=0x378,7 hdd=ide-scsi"' wordt de kernel verteld om voor de parallelle poort het adres 0x378 en interrupt 7 te gebruiken en dat 'hdd' mijn CD brander is die via SCSI-emulatie aangestuurd dient te worden ('hdd' is systeem-specifiek; op uw systeem kan dat dus anders zijn). Het gebruik van een interrupt maakt het printen wat sneller (standaard gebruikt de Linux kernel het tragere 'polling'). Als u geen interrupt meer vrij hebt voor de parallelle (printer)poort, dan laat u deze regel weg. 'Polling' betekent dat de kernel het gebruik van een interrupt omzeilt, waardoor de parallelle poort toch gebruikt kan worden. Deze instelling is een kwestie van persoonlijke voorkeur. De laatste sectie (vanaf 'image = /boot/vmlinuz.suse') bevat de lilo-instelling zoals die door SuSE tijdens de configuratie is ingesteld (de regel append="hdd=ide-scsi" heb ik wel handmatig toegevoegd). Het bestand 'boot/vmlinuz.suse' is de standaard kernel die bij de distributie is geleverd en die u bij voorkeur ALTIJD bewaart voor eventuele noodgevallen! De regel 'initrd = /boot/initrd.suse' is alleen van toepassing op de standaard geïnstalleerde Linux kernel en regelt het inladen van een 'ramdisk' image: een virtuele schijf die in het geheugen wordt ingeladen. De 'ramdisk' bevat modules die noodzakelijk

zijn voor een correct opstarten van Linux. Een distributiekernel moet nou eenmaal ontzettend veel verschillende soorten hardware kunnen aansturen, wat alleen kan door veel modules te gebruiken.

Hopelijk is u nu duidelijk waar u de nieuwe Linux kernel in dit voorbeeld naartoe moet kopiëren:

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot
cp /usr/src/linux/System.map /boot/System.map-2.4.6
lilo
```

Onder SuSE 7.3 kan de tweede kopieeropdracht ook als volgt uitgevoerd worden:

```
cp /usr/src/linux/System.map /boot
```

(de originele System.map is reeds hernoemd)

Als u reeds een kernel met de naam 'bzImage' heeft en u wilt die kernel bewaren, dan kopieert u de nieuwe kernel als '/boot/bzImage-2.4.6' en zet u deze wijziging ook in '/etc/lilo/conf'. Het bestand 'System.map' wordt tijdens de compilatie altijd gecreëerd en bevat de namen en instellingen van belangrijke kernel variabelen. De opdracht 'depmod -a' zorgt ervoor dat een bestand wordt aangemaakt, waarin alle afhankelijkheden staan van de kernel modules (relaties tussen kernel en module en modules onderling). Ook de informatie in '/etc/modules.conf' wordt meegenomen. De meeste Linux distributies (inclusief SuSE) zorgen ervoor dat tijdens het opstarten 'depmod -a' wordt gedraaid, maar voor de zekerheid is het verstandig om ervoor te zorgen dat het bestand /boot/System.map bestaat (en overeenkomt met de huidige kernelversie). De opdracht 'lilo' moet u nog uitvoeren om de nieuwe instellingen (of de nieuwe kernel) effectief te installeren. Als u reeds een bestand '/boot/bzImage' hebt en u overschrijft dit bestand met een nieuw gecompileerde kernel zonder 'lilo' te draaien, dan kan de nieuwe kernel niet meer opstarten. De oude (originele) distributiekernel die u hebt bewaard werkt dan nog wel, want die is immers niet overschreven! Op deze wijze kunt u op een verantwoorde wijze nieuwe kernels compileren en testen...

Loadlin gebruiken

Als u 'loadlin' gebruikt om Linux op te starten, dan kent u ongetwijfeld de lokatie van dit handige programmaatje, bijvoorbeeld C:\loadlin. 'Loadlin' vindt u op de eerste CD in de 'dosutils' directory van nagenoeg iedere distributie. De nieuwe Linux kernel dient u ook naar de harde schijf ('C:\loadlin') te kopiëren; eventueel met een unieke naam. Met de opdracht:

loadlin bzImage

wordt Linux vervolgens opgestart onder DOS (bijvoorbeeld door Windows98 eerst op te starten in DOS modus). Als u de kernel zelf hebt gecompileerd, dan zijn de meeste instellingen zoals die in de kernel zelf zijn opgeslagen (zoals de locatie van de root partitie) normaal gesproken correct. Bovenstaande opdracht volstaat dan om probleemloos Linux op te starten. Als u in DOS gewoon de opdracht 'loadlin |more' intypt, dan krijgt u een helpscherm inclusief een link naar een 'loadlin-HowTo' op internet. Als u huiverig bent om met 'lilo' te stoeien, dan kunt u op deze wijze zonder risico de nieuw gecompileerde Linux kernel uitproberen. 'Loadlin' en 'bzImage' passen namelijk samen op een diskette (1.44 MB). Als u eerst DOS opstart via een opstartdiskette (met EMM386 ondersteuning) en daarna de 'loadlin+bzImage' diskette gebruikt om Linux op te starten, dan kan er werkelijk helemaal niets foutgaan! Uiteraard dient u wel een DOS opstartdiskette te hebben.

SuSE en de ALSA geluidsdrivers

SuSE gebruikt standaard de ALSA geluidsdrivers. Deze drivers zijn van betere kwaliteit dan de drivers van het OSS-project die standaard in de kernel zitten. Als u echt met geluid aan de slag gaat onder Linux, dan is het gebruik van de ALSA-drivers een "must"! De ALSA-drivers zitten niet bij de kernel broncode meegeleverd. Dit betekent dat de drivers apart gecompileerd en geïnstalleerd dienen te worden. De ALSA-drivers zitten in de ALSA broncode in het onderdeel 'zq' van de SuSE distributie (als dit u niets zegt, lees dan even verder). Installeer ALSA via YaST of YaST2 zodat u de broncode beschikbaar hebt in de directory '/usr/src/packages/'. Om de ALSA drivers te compileren en te installeren gaat u als volgt te werk:

```
rpm -bb /usr/src/packages/SPECS/alsa.spec
cd /usr/src/packages/BUILD/alsa/alsa-driver-<versienummer>/
./configure
make install
```

De bovenste regel installeert de broncode, inclusief de drivers, in de '/usr/src/packages/BUILD/' directory. Bovendien worden de ALSA libraries en utilities meteen als rpm-bestand gecompileerd! Helaas worden de ALSA drivers NIET standaard meegecompileerd. U dient deze zelf met de hand te compileren en te installeren. De onderste twee opdrachten zorgen daarvoor. De './configure' opdracht zoekt de instellingen en bestandslocaties van uw systeem op en zet deze in een configuratiebestand. De opdracht 'make install' compileert alle ALSA drivers en installeert deze meteen voor de kernel (in de '/lib/modules/2.4.6/misc/' directory. Tijdens het opstarten van SuSE wordt nu automatisch de juiste geluidsdriver geïnstalleerd. Ik geef toe dat bovenstaande werkwijze enigszins omslachtig is. Alleen mensen die weten waar ze mee bezig zijn, kunnen de drivers vinden en de geluidsondersteuning in een nieuwe kernel inbrengen.

Als u geen SuSE gebruikt of als u een recentere versie van de ALSA drivers wilt gebruiken, dan kunt u deze drivers (samen met de 'libraries' en de 'utilities') downloaden van <http://www.alsa-project.org>. De startpagina van deze site bevat meteen links naar de verschillende te downloaden onderdelen. Ik ga nu uit van de compilatie van de ALSA drivers, maar de beschreven stappen zijn identiek voor de 'libraries' en de 'utilities.' Pak de drivers uit in een directory, bijvoorbeeld in '/usr/local/'. Ga naar deze directory (bijvoorbeeld '/usr/local/alsa-driver-<versienummer>/' en voer vervolgens de bovenvermelde stappen uit (beginnend met './configure'). Als uw distributie niet standaard gebruik maakt van de ALSA drivers, kan het mogelijk zijn dat u wat extra stappen moet ondernemen om de drivers correct werkend te krijgen. Helaas valt dit buiten het bestek van dit al zeer uitvoerige artikel, maar de ALSA FAQ (Frequently Asked Questions) die u apart kunt downloaden helpt u verder op weg.

pcmcia ondersteuning (laptops)

Vanaf kernel 2.4.x is er ook pcmcia ondersteuning in de kernel. Echter, het gebruik van de kernel drivers wordt (in de officiële HOWTO) vooralsnog afgeraden. De komende tijd heeft het pcmcia (source) pakket zowel drivers als scripts om de pcmcia kaart werkend te krijgen voor zowel 2.0.x, 2.2.x en 2.4.x kernels. Mijn ervaring is dat na een kernel compilatie de pcmcia drivers niet meer werken, waardoor een

hercompilatie noodzakelijk is. Afhankelijk van de distributie die gebruikt wordt, leiden er twee wegen naar Rome. De eerste methode is de broncode die bij de distributie meegeleverd is. Het installeren van de broncode en het compileren daarvan naar een rpm-bestand levert u een bestand op dat u kunt installeren. De andere methode is het downloaden, uitpakken, compileren en installeren van de meest recente versie van pcmcia (die kunt u vinden op <http://sourceforge.net/projects/pcmcia-cs/>):

```
cp /etc/rc.d/pcmcia /etc/rc.d/pcmcia.SuSE
cp ~/pcmcia-cs-3.1.?.tar.gz /usr/src
cd /usr/src
tar -zxf ./pcmcia-cs-3.1.?.tar.gz
make config
make all
make install
cp /etc/rc.d/pcmcia.SuSE /etc/rc.d/pcmcia
```

De eerste en laatste regels lossen een probleem op onder SuSE. Het SuSE-initialisatiescript `/etc/rc.d/pcmcia` wordt overschreven door de `'make install'` opdracht, waardoor het script onder SuSE niet meer correct functioneert. Door het originele script terug te kopiëren wordt het probleem opgelost. Mocht u onverhoopt toch het originele SuSE script overschreven hebben, dan dient u het pakket `'pcmcia'` opnieuw te installeren uit onderdeel `'a1'`, het originele script te kopiëren naar een ander bestand, `'make install'` opnieuw uit te voeren en het originele script terug te plaatsen.

Het nieuwe rpm-bestand voor pcmcia ondersteuning installeert u onder SuSE als volgt:

```
rpm -i /cdrom/suse/zq1/pcmcia-3.1.?.spm
cd /usr/src/packages
rpm -bb ./SPECS/pcmcia-3.1.?.spec
cd /RPMS/i386/
rpm -i --force ./pcmcia-3.1.?.rpm
SuSEconfig
```

In de eerste regel ga ik ervan uit dat u de pcmcia-drivers zelfstandig vanaf de 6e of 7e CD installeert en dat de CDROM speler reeds gemount is via `/cdrom`. De `'rpm -i'` opdracht installeert de broncode en de `'rpm -bb'` opdracht compileert het pcmcia rpm-bestand. Vervolgens installeert u dit rpm pakket met de hand zoals elk gewoon rpm-bestand. Merk op dat u wel de optie `'--force'` moet meegeven omdat het programma `'rpm'` anders terecht opmerkt dat `'pcmcia'` reeds is geïnstalleerd (en het nieuwe bestand dus negeert). Zoals altijd dient u na het handmatig installeren van rpm-bestanden onder SuSE het programma `'SuSEconfig'` (let op de hoofd- en kleine letters!) te draaien om eventuele gewijzigde instellingen te activeren. Dit laatste wordt door het SuSE setup programma YaST/YaST2 altijd automatisch uitgevoerd na het installeren of wijzigen van nieuwe pakketten, waardoor u dit normaal gesproken niet handmatig hoeft te doen.

Om pcmcia support goed te kunnen gebruiken, dient tijdens de kernelcompilatie `'network support'` aan te staan, maar moeten alle overige netwerkkaartdrivers uitgeschakeld zijn. Als u wilt internetten dient u uiteraard ook `'TCP/IP support'` aan te zetten, zoals al eerder is behandeld (zie Figuur 11).

Het installeren van de `'modutils'`

Zoals eerder vermeld bevatten de 'modutils' kleine programmaatjes die de kernel nodig heeft om de kernel-modules te kunnen beheren. Dit zijn dan programma's als:

```
insmod ( installeren van een module           ),
rmmmod ( verwijderen van een module          ),
lsmod  ( alle in gebruik zijnde modules tonen ),
```

om maar een paar voorbeelden te noemen. Met de opdracht 'man lsmod' kunt u bijvoorbeeld kijken hoe deze opdracht nou precies werkt. Daar ga ik nu niet verder op in.

Het compileren en installeren van de 'modutils' is rechttoe-rechtaan:

```
cd /usr/src
cp ~/modutils-2.4.6.tar.bz2 . ( ervan uitgaande dat desbetreffend      )
                             ( bestand in uw home-directory ('~') staat )
bzip2 -d modutils-2.4.6.tar.bz2 ( dit kan even duren                    )
tar -xvf modutils-2.4.6.tar
cd modutils-2.4.6              ( ga naar de directory waar de 'modutils'  )
                             ( net zijn uitgepakt                       )
./configure                    ( zoek systeemspecifieke instellingen op )
make ( compileer de 'modutils'. Omdat het erg klein is, gaat dit       )
    ( verrassend snel                                                  )
make install ( installeer de 'modutils' in de directory '/sbin/'        )
```

Merk opnieuw op dat het in dit voorbeeld toevallig is dat de 'modutils' en de kernel dezelfde versie nummers hebben. Dit is alles wat u dient te doen. De nieuwe 'modutils' zijn klaar voor gebruik.

Werkt de kernel correct?

De nieuwe kernel is geconfigureerd, gecompileerd en waarschijnlijk ook geïnstalleerd via 'lilo'. U herstart het systeem en vraagt zich nu af hoe te controleren of de nieuw gecompileerde kernel goed functioneert. Als bepaalde hardware niet werkt, komt u daar snel genoeg achter tijdens het uitproberen. Echter, tijdens het opstarten drukt de kernel veel nuttige informatie af op het beeldscherm (voordat bijvoorbeeld de grafische login wordt opgestart). Deze informatie bevat o.a. automatisch gedetecteerde instellingen (poorten, IRQs, en dergelijke), maar ook foutmeldingen, bijvoorbeeld als een bepaalde driver (als module of als kernelonderdeel) niet correct geïnitieerd kan worden. Door tijdens het opstarten goed op te letten, kunt u bepaalde problemen al snel detecteren. Bij SuSE linux gaat dit zelfs erg makkelijk: ieder 'kernelonderdeel' dat opgestart wordt krijgt uiterst rechts op het beeldscherm de melding 'done' (in het groen) als alles correct is verlopen of 'failed' (in het rood) als er ergens iets is misgegaan. Omdat deze meldingen erg snel over het scherm kunnen scrollen, worden eventuele foutmeldingen nog kort samengevat boven de 'login-prompt'. Deze login-prompt vindt u onder '<Ctrl>+<Alt>+F1' als u automatisch het grafische inlogscherf krijgt. U hebt nu in ieder geval een hint waar eventuele problemen zich kunnen voordoen, vooropgesteld dat de foutmelding voor u al relevant is. Foutmeldingen die lijken op 'Cannot find module' of 'Cannot load module' duiden er meestal op dat u tijdens de kernelcompilatie enkele onderdelen niet hebt meegenomen die u wel nodig hebt. Even de kernelconfiguratie aanpassen en de kernel opnieuw compileren en installeren lost het probleem meestal vanzelf op. Merk op dat het niet noodzakelijk hoeft te zijn om de hele kernel te compileren. Als u alleen

wat modules bent vergeten, volstaat 'make modules' en 'make modules_install'. Mocht u onverhoopt toch de hele kernel opnieuw moeten compileren, dan is er nog niet veel aan de hand. De meeste kernel source code is reeds in gecompileerde vorm aanwezig en alleen ontbrekende onderdelen hoeven opnieuw gecompileerd te worden. Kortom: de kernel iets aanpassen en opnieuw compileren gaat heel snel (in tegenstelling tot de eerste kernelcompilatie; dan mag u rustig een kopje koffie gaan drinken).

Er bestaat ook nog een aparte opdracht om de kernel 'boot messages' te kunnen bekijken, namelijk de opdracht 'dmesg'. Als u informatie nodig heeft die reeds van het scherm is 'afgescrold', dan kunt u met deze opdracht de 'kernel messages' ook nog nalezen. U moet wel de volgende opdracht gebruiken: 'dmesg > temp', waarmee alles in het bestand 'temp' wordt opgeslagen. Met de opdracht 'more temp' kan U het bestand vervolgens lezen.

Tot slot

Met deze 'handleiding' kunt u op verantwoorde wijze gaan experimenteren met kernel compilatie. De drempel om zelf met de kernel te gaan 'stoeien' is hopelijk zodanig verlaagd dat u enthousiast aan de slag gaat. Het correct configureren van de kernel vergt nog de meeste tijd. Tijdens het compileren kunt u rustig een spelletje 'patience' spelen of andere computerwerkzaamheden verrichten.

Mocht u onverhoopt toch nog op onoverkomelijke problemen stuiten, dan bestaan er op internet een aantal Nederlandstalige Linux forums waarop u vragen kunt stellen. Normaal gesproken krijgt u vrij snel een zinnig antwoord terug waarmee u het probleem definitief kunt oplossen. Via een zoekmachine kunt u deze forums snel vinden.

Site onderhouden door het LinuxFocus editors
team

© Philip de Groot

"some rights reserved" see linuxfocus.org/license/
<http://www.LinuxFocus.org>

Vertaling info:

nl --> -- : Philip de Groot <philipg/at/authors.linuxfocus.org>