De Linux Kernel HOWTO

Brian Ward, bri@cs.uchicago.edu,
Vertaald door: Ellen Bokhorst, bokkie@nl.linux.org v2.6, 18 mei 2001

Dit is een gedetailleerde gids voor het configureren, compileren en upgraden van de kernel, en het oplossen van problemen voor op ix86 gebaseerde systemen.

Inhoudsopgave

1 Introductie

Zou je dit document moeten lezen? Wel als je één van de volgende symptomen herkent:

- "Ai! Dit wizzo-46.5.6 package zegt dat het kernel release 2.8.193 nodig heeft en ik heb nog steeds slechts release 1.0.9!"
- Er is een devicedriver in één van de nieuwere kernels die je gewoon moet hebben
- Je hebt werkelijk geen idee hoe je een kernel compileert
- "Is datgene wat er in de README staat, *echt* het hele verhaal?"
- Je kwam, je probeerde het, het werkte niet
- Je hebt iets nodig om aan mensen te geven die erop blijven hameren dat je hun kernels voor ze installeert

1.1 Lees dit eerst! (Ik meen het)

Bij een aantal van de voorbeelden in dit document wordt ervan uitgegaan dat je GNU `tar`, `find`, en `xargs` hebt. Dit zijn standaards; dit zou geen problemen mogen veroorzaken. Er wordt ook vanuit gegaan dat je bekend bent met de structuur van het bestandssysteem op je systeem; als dit niet zo is, is het van groot belang dat je een geschreven kopie van de uitvoer van het commando `mount` tijdens de normale systeembewerking bij de hand houdt (of een listing van `/etc/fstab`, als je het kunt lezen). Deze informatie is belangrijk, en wijzigt niet, tenzij je je disk opnieuw partitioneert, een nieuwe disk toevoegt, je systeem opnieuw installeert of iets dergelijks.

De laatste “productie” kernelversie op moment van dit schrijven was 2.2.9, wat betekent dat de verwijzingen en voorbeelden corresponderen met die release. Zelfs al probeer ik dit document zo versie-onafhankelijk mogelijk te maken, de kernel is voortdurend onder ontwikkeling, dus als je een nieuwere release krijgt, zullen er onontkoombaar wat verschillen zijn. Nogmaals, dit zou geen belangrijke problemen moeten geven, maar het kan wat verwarring geven.

Er zijn twee versies van de linux kernel source, “productie” en “development.” Productie releases zijn de even-minor-genummerde releases; 1.2.x was productie, 2.0.x is productie, als ook 2.2.x. Deze kernels worden aangemerkt als de meest stabiele kernels, foutvrije versies die ten tijde van de release beschikbaar zijn. De development kernels (2.1.x, 2.3.x, enz) zijn bedoeld als kernels om te testen, voor mensen die bereid zijn om ze uit te testen en het kunnen kernels zijn die veel fouten bevatten. Je bent gewaarschuwd.

1.2 Een woord over stijl

`Tekst die er zo uitziet` is iets dat op het scherm verschijnt, een bestandsnaam, of iets dat direct kan worden ingetikt, zoals een commando, of opties voor een commando (als je een plain-tekstbestand bekijkt, ziet het er niet veel anders uit). Commando’s en andere invoer worden vaak tussen aanhalingstekens geplaatst (met ‘ ’), wat het volgende klassieke leesteken-probleem veroorzaakt: als een dergelijk item aan het einde van een zin tussen aanhalingstekens verschijnt, typen mensen vaak een ‘.’ met het commando, omdat de Amerikaanse aanhalingsstijl aangeeft dat de punt binnen de aanhalingstekens hoort. Zelfs met gezond verstand (en helaas wordt hier verondersteld dat degene met “gezond verstand” gewend is aan de zogenoemde Amerikaanse stijl van aanhalen), zou iemand moeten zeggen het eerst van het leesteken te ontdoen, toch denken veel mensen daar gewoon niet aan, dus ik zal het in dergelijke gevallen buiten de aanhalingstekens plaatsen. Met andere woorden, als er wordt aangegeven dat je “`make config`” zou moeten typen, zal ik ‘`make config`’ schrijven en niet ‘`make config.`’

2 Snelle stappen - Kernelcompilatie

Deze sectie is geschreven door *Al Dev (alavoor@yahoo.com)*

De laatste versie van deze sectie is te vinden op <http://www.aldev.8m.com> en klik op "Quick Steps to recompile linux kernel". Mirror sites zijn te vinden op - <http://aldev.webjump.com>, *angelfire* <http://www.angelfire.com/country/aldev0>, *geocities* <http://www.geocities.com/alavoor/index.html>, *virtualave* <http://aldev0.virtualave.net>, *50megs* <http://aldev0.50megs.com>, *theglobe* <http://members.theglobe.com/aldev1/index.html>, *NBCi* <http://members.nbci.com/alavoor>, *Terrashare* <http://aldev.terrashare.com>, *Fortunecity* <http://members.fortunecity.com/aldev>, *Freewebsites* <http://aldev.freewebsites.com>, *Tripod* <http://members.tripod.lycos.com/aldev>, *Spree* <http://members.spree.com/technology/aldev>, *Escalix* <http://www.escalix.com/freepage/aldev>, *Httpcity* <http://www.httpcity.com/aldev/index.html>, *Freeservers* <http://aldev.freeservers.com>.

Deze sites bieden **heel veel linux fraais** en tips.

Een kopie van de bovenstaande website is hier gereproduceerd -

Het opnieuw compileren van de kernel is nodig om de kernel zo klein mogelijk te maken, wat zal resulteren in een SNELLER besturingssysteem. Het moet ook gebeuren als je ondersteuning wilt voor enige nieuwe devices. **Noot:** Hieronder duidt ’bash#’ op de bash-prompt, je typt de opdrachten die na de ’bash#’ prompt staan. De hieronderstaande opdrachten zijn onder RedHat Linux getest, maar ze zouden tevens met minimale wijzigingen onder andere distributies moeten werken.

1. Noot: Er kunnen meerdere kernelimages op je systeem voorkomen. Door de hieronder staande stappen op te volgen, overschrijf je of beschadig je geen bestaande kernel. Deze stappen zijn **zeer veilig** en je

huidige kernel blijft intact en zal niet worden aangeroerd.

2. Log bij al deze stappen in als ’root’. Mount de Redhat linux cdrom en installeer de linux kernelsource rpm

```
bash$ su - root
bash# cd /mnt/cdrom/RedHat/RPMS
bash# rpm -i    kernel-headers*.rpm
bash# rpm -i    kernel-source*.rpm
bash# rpm -i dev86*.rpm
bash# rpm -i bin86*.rpm
```

(De bin86*.rpm en ’as86’ is alleen nodig voor **OUDERE Linux** systemen zoals redhat 5.x. Neem de Intel assembler ’as86’ opdracht uit dev86*.rpm op de cdrom of vanaf `<http://rpmfind.net/linux/RPM/mandrake/7.1/Mandrake/RPMS/bin86-0.4-12mdk.i586.html>` , `<http://rpmfind.net/linux/RPM/kondara/jirai/i586/bin86-0.4-8k.i586.html>`).

3. Start X-window op met ’startx’.

```
bash# man startx
bash# startx
bash# cd /usr/src/linux
bash# make xconfig
```

De **"make xconfig"**

komt met een gebruikersvriendelijke GUI interface! Maak **GEEN** gebruik van ’make config’ want dit is een opdrachtregel optie (gebruik dit **ALLEEN** als je X-window niet kunt uitvoeren). Een beter alternatief op de opdracht ’make config’ is **’make menuconfig’** waarmee je een gebruikersvriendelijke op tekst gebaseerde ncurses interface krijgt. Een aantal mensen geven de voorkeur aan ’make menuconfig’ boven ’make xconfig’. Mijn persoonlijke voorkeur - wanneer ik X-window niet uit kan voeren, gebruik ik ’make menuconfig’. Je kunt het configuratiebestand laden van */usr/src/linux/.config* (punt config file. Let op de punt voor config).

4. Binnen ’make xconfig’ moet je het volgende doen om problemen te voorkomen -

   - Selecteer het juiste CPU type - Pentium 3, AMD K6, Cyrix, Pentium 4, Intel 386, DEC Alpha, PowerPC anders zal de kernel niet booten!!
   - Kies voor SMP - of een enkele CPU of meerdere CPU’s
   - Filesystems - Selecteer Windows95 Vfat, MSDOS, NTFS als onderdeel van de kernel en niet als laadbare modules.
   - Activeer de ondersteuning voor laadbare kernelmodules! Met deze optie kun je device drivers op een draaiend linux systeem dynamisch laden en uit het geheugen verwijderen. Zie de volgende manpages

```
bash# man lsmod
bash# man insmod
bash# man rmmod
bash# man depmod
bash# man modprobe
```

5. Save en Exit "make xconfig". Alle opties die je hebt geselecteerd zullen nu worden opgeslagen in het configuratiebestand */usr/src/linux/.config* (punt config file). En voer nu uit:

```
bash# make dep
bash# make clean
```

---

6. Lees het volgende bestand (om kennis op te doen over het bouwen van een kernel. Tip: gebruik de kleureneditor *gvim* `<http://metalab.unc.edu/LDP/HOWTO/Vim-HOWTO.html>` voor een betere leesbaarheid.

```
bash# gvim -R   /usr/src/linux/arch/i386/config.in
bash# man less
bash# less   /usr/src/linux/arch/i386/config.in
Typ 'h' voor hulp en druk voor het navigeren op i, j, k, l, h of
de pijltjestoets, page up/down toetsen.
```

---

7. Geef nu de opdracht make:

```
        bash# cd /usr/src/linux
        bash# man nohup
        bash# nohup make bzImage &
        bash# tail -f nohup.out     (.... to monitor the progress)
Hiermee zal de kernel worden neergezet in
/usr/src/linux/arch/i386/boot/bzImage
        bash# man tail
```

---

8. Na een succesvolle bzImage, kopieer je de kernelimage naar de /boot directory. Je moet de nieuwe kernelimage naar de /boot directory kopiëren, anders zal de nieuwe kernel **NIET** booten. En lees dan de manual page over lilo (zie ook `<http://www.linuxdoc.org/HOWTO/LILO-crash-rescue-HOWTO.html>`) en zie het voorbeeldbestand 2.2 (lilo.conf). Geef altijd een datumextensie aan de bestandsnaam, omdat het je vertelt wanneer je de kernel bouwde. Dit wordt hieronder aangegeven:

```
bash# cp /usr/src/linux/arch/i386/boot/bzImage     /boot/bzImage.myker.26mar2001
bash# man lilo
bash# man lilo.conf
Wijzig het bestand /etc/lilo.conf en plaats daar de volgende regels in:
        image=/boot/bzImage.myker.26mar2001
        label=myker
        root=/dev/hda1
        read-only
Je kunt de devicenaam voro 'root=' controleren met de opdracht:
        bash# df   /
```

---

9. Start nu

```
bash# lilo
bash# lilo -q
```

Je moet lilo iedere keer dat je een nieuwe bzImage aanmaakt, opnieuw opstarten, zelfs als zou het record 'myker' reeds bestaan.

10. Start de machine opnieuw op en achter lilo druk je de tab toets in en typt 'myker'. Als het boot dan heb je goed werk verricht! Selecteer anders achter de lilo je oude kernel, boot en begin helemaal opnieuw. Je oude kernel **is nog steeds INTACT en VEILIG gesteld** in stel */boot/vmlinuz-2.0.34-0.6*

11. Laadbare modules: Zoek naar de insmod opdracht die zeer frequent wordt gebruikt voor het laden van de modules.

```
 bash# man insmod
 bash# insmod
 bash# rpm -i /mnt/cdrom/Redhat/RPMS/modutils*.rpm
```

De stap hieronder is wellicht niet nodig, maar slechts nodig voor **NOODGEVALLEN** wanneer je /lib/modules bestanden zijn beschadigd. Wanneer je reeds een /lib/modules directory hebt en in het geval je de bestanden wilt vervangen, gebruik dan de –force om het package te vervangen en selecteer de van toepassing zijnde cpu architectuur.

Voor nieuwe versies van linux redhat linux 6.0 en latere versies zijn de kernelmodules opgenomen in kernel-2.2*.rpm. Installeer de laadbare modules en de kernel met

```
Hiermee krijg je een reeds geïnstalleerd package te zien.
 bash# rpm -qa | grep -i kernel

 bash# rpm -U --force  /mnt/cdrom/Redhat/RPMS/kernel-2.2.14-5.0.i686.rpm
 (or)
 bash# rpm -U --force  /mnt/cdrom/Redhat/RPMS/kernel-2.2.14-5.0.i586.rpm
 (or)
 bash# rpm -U --force  /mnt/cdrom/Redhat/RPMS/kernel-2.2.14-5.0.i386.rpm
```

Dit geldt alleen voor oude versies van redhat linux 5.2 en eerdere versies. Boot de nieuwe kernel en installeer de laadbare modules vanaf de RedHat Linux "contrib"cdrom

```
bash# rpm -i /mnt/cdrom/contrib/kernel-modules*.rpm
....(Voor oude linux systemen waarop insmod niet is voorgeïnstalleerd)
```

12. Deze stap is **ALLEEN** nodig als je een nieuwe versie van de kernelsource hebt gedownload. Laadbare modules zijn te vinden in /lib/modules.

```
bash# cd /usr/src/linux
bash# make modules
bash# make modules_install
```

13. Als je nieuwe kernel 'myker' boot en correct functioneert, dan kun je de bootdisk aanmaken. Doe een lege diskette in het diskettestation en:

```
bash# cd /usr/src/linux
bash# make bzdisk
Zie tevens mkbootdisk -
bash# rpm -i mkbootdisk*.rpm
bash# man mkbootdisk
```

2.1 Oplossen van veelvoorkomende fouten

De volgende fouten worden zeer frequent door nieuwe gebruikers gemaakt:

Als je nieuwe kernel niet boot en je krijgt:

```
Warning: unable to open an initial console
Kernel panic: no init found. Try passing init= option to kernel
```

Het probleem is hier dat je de "root="parameter **NIET** juist opgaf in /etc/lilo.conf. In mijn geval maakte ik gebruik van root=/dev/hda1 met daarop de root partitie /". Je moet naar het juiste rootdevice verwijzen in lilo.conf. Dit kan bv iets als /dev/hdb2 of /dev/hda7 zijn.

De kernel zoekt naar de init opdracht welke is te vinden in /sbin/init. En de directory /sbin komt voor op de rootpartitie. Zie voor details:

```
bash# man init
```

2.2 Voorbeeld lilo.conf

Hieronder wordt een voorbeeldbestand van /etc/lilo.conf gegeven. Volg de naamconventies zoals ker2217 (voor kernel 2.2.17), ker2214 (voor kernel 2.2.14). Er kunnen meerdere kernelimages op hetzelfde /boot systeem voorkomen. Op mijn machine heb ik iets als:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=firewall

image=/boot/vmlinuz-2.2.14-5.0
       label=ker2214
       read-only
       root=/dev/hda9

image=/boot/vmlinuz-2.2.17-14
       label=ker2217
       read-only
       root=/dev/hda9

#image=/usr/src/linux/arch/i386/boot/bzImage
#      label=myker
#      root=/dev/hda7
#      read-only

image=/boot/bzImage.myker.11feb2001
       label=myker11feb
       root=/dev/hda9
       read-only
```

```
image=/boot/bzImage.myker.01jan2001
        label=myker01jan
        root=/dev/hda9
        read-only

image=/boot/bzImage.myker-firewall.16mar2001
        label=firewall
        root=/dev/hda9
        read-only
```

3 Belangrijke vragen en de antwoorden daarop

3.1 Wat doet de kernel, eigenlijk?

De Unix kernel gedraagt zich als een bemiddelaar voor je programma's en je hardware. Als eerste doet het (of regelt het) voor alle draaiende programma's het geheugenbeheer (processen), en zorgt het ervoor dat ze allen een eerlijk (of oneerlijk, als je dat wenst) deel van de processor cycli krijgen. Bovendien voorziet het in een tamelijk fraaie overdraagbare interface voor programma's om met je hardware te communiceren.

De werking van de kernel beslaat beslist meer dan dat, maar dit zijn de belangrijkste basisfuncties.

3.2 Waarom zou ik mijn kernel willen upgraden?

Nieuwere kernels bieden over het algemeen de mogelijkheid om te kunnen communiceren met meer soorten hardware (dat wil zeggen dat ze meer devicedrivers hebben), het kan zijn dat ze een beter procesbeheer hebben, ze kunnen sneller draaien dan de oudere versie, ze zouden stabieler kunnen zijn dan de oudere versies, en ze herstellen domme fouten in de oudere versies. De meeste mensen upgraden kernels omdat ze de devicedrivers en de bug fixes willen.

3.3 Welke soort hardware ondersteunen de nieuwere kernels?

Zie de Hardware-HOWTO. Als alternatief kun je het bestand '`config.in`' in de linux source bekijken, of er gewoon achterkomen als je '`make config`' uitprobeert. Hierdoor wordt alle door de standaard kerneldistributie ondersteunde hardware getoond, maar niet alles dat door linux wordt ondersteund; veel algemene devicedrivers (zoals de PCMCIA drivers en een aantal tape drivers) bestaan uit laadbare modules die apart worden beheerd en gedistribueerd.

3.4 Welke versie van gcc en libc heb ik nodig?

Linus beveelt in het bestand `README`, dat bij de linux source is inbegrepen, een gcc-versie aan. Als je deze versie niet hebt, zou de documentatie van de aanbevolen versie van gcc aan moeten geven of je je libc moet upgraden. Dit is geen moeilijke procedure, maar het is belangrijk de instructies op te volgen.

3.5 Wat is een laadbare module?

Dit zijn delen van de kernelcode die niet direct in de kernel zijn gelinkt (ingevoegd). Ze worden apart gecompileerd, en je kunt ze op bijna ieder moment in de draaiende kernel invoegen en verwijderen. Vanwege de flexibiliteit ervan, is dit nu de te verkiezen manier om bepaalde kernelfaciliteiten te coderen. Veel van de populaire devicedrivers, zoals de PCMCIA drivers en de QIC-80/40 tapedriver, zijn laadbare modules.

3.6 Hoeveel diskruimte heb ik nodig?

Dat is afhankelijk van je afzonderlijke systeemconfiguratie. Ten eerste is de gecomprimeerde linux-source van versie 2.2.9 bijna 14 megabytes groot. Veel sites houden dit zelfs na het uitpakken. Ongecomprimeerd en gebouwd met een gemiddelde configuratie, neemt het nog eens 67 MB in beslag.

3.7 Hoeveel tijd neemt het in beslag?

Met nieuwere computers, neemt de compilatie aanmerkelijk minder tijd in beslag dan met oudere computers; een AMD K6-2/300 met een snelle harddisk kan een 2.2.x kernel in ongeveer vier minuten compileren. Als je van plan bent om te gaan compileren, wees dan bij oude Pentiums, 486'rs, en 386'rs voorbereid dat je zult moeten wachten, mogelijk wel uren, dagen..

Als je je daar zorgen over maakt, en je hebt een snellere computer in de buurt om op te compileren, kun je op de snelle computers bouwen (ervan uitgaande dat je het de juiste parameters meegeeft, dat je ulilities up-to-date zijn, enzovoort), en vervolgens de kernel-image naar de langzamere computer transporteren.

4 Hoe de kernel feitelijk te configureren

4.1 Verkrijgen van de source

Je kunt de source via anonieme ftp verkrijgen vanaf `ftp.kernel.org` in `/pub/linux/kernel/vx.y`, waar `x.y` voor de versie staat (bv 2.2), en zoals eerder genoemd, die met een oneven nummer eindigen, zijn development releases en kunnen onstabiel zijn. Ze zijn vaak gelabeld als `linux-x.y.z.tar.gz`, waarbij `x.y.z` voor het versienummer staat. Op de sites staan vaak ook bestanden met het toevoegsel `.bz2`, deze zijn met bzip2 gecomprimeerd (ze zullen kleiner zijn en nemen bij het transporteren minder tijd in beslag).

Je kunt het beste `ftp.xx.kernel.org` gebruiken, waar `xx` je landcode voorstelt; voorbeelden zijn `ftp.at.kernel.org` voor Australië, en `ftp.us.kernel.org` voor de United States.

4.2 Het uitpakken van de source

Log in als, of `su` naar, '`root`', en `cd` naar `/usr/src`. Als je de kernelsource installeerde toen je linux voor het eerst installeerde (zoals de meeste doen), zal er reeds een directory met de naam '`linux`' voorkomen, die de gehele oude sourcetree bevat. Als je de diskruimte hebt en het veilig wilt spelen, bewaar die directory dan. Het is een goed idee om er achter te komen welke versie nu op je systeem draait en de directory overeenkomstig te hernoemen. Het commando '`uname -r`' drukt de huidige kernelversie af. Daarom zou je (met '`mv`') '`linux`' in '`linux-1.0.9`' kunnen hernoemen, als '`uname -r` '`1.0.9`' aangeeft. Als je je daar niet zo om bekommert, verwijder dan gewoon de volledige directory. Zorg er in ieder geval voor dat er geen '`linux`' directory in `/usr/src` voorkomt, voordat je de volledige sourcecode uitpakt.

Pak nu in `/usr/src` de source uit met '`tar zxpvf linux-x.y.z.tar.gz`' (als je slechts een `.tar` bestand zonder `.gz` aan het einde hebt, werkt '`tar xpvf linux-x.y.z.tar`'). De inhoud van de source zal voorbij vliegen. Als 't klaar is, dan zal er een nieuwe '`linux`' directory in `/usr/src` voorkomen. `cd` naar `linux` en lees het bestand `README` door. Er zal een sectie in staan met het label '`INSTALLING the kernel`'. Voer de van toepassing zijnde instructies uit, symbolische links die op hun plaats zouden moeten staan, verwijdering van oude `.o` bestanden, enz.

Als je een `.bz2` bestand en het bzip2 programma hebt (lees erover op `http://www.muraroa.demon.co.uk/`), doe dan het volgende:

```
bz2cat linux-x.y.z.tar.bz2 | tar xvf -
```

4.3 Configureren van de kernel

Opmerking: Een deel hiervan is een herhaling/opheldering van een vergelijkbare sectie in het `README` bestand van Linux.

Het commando '`make config`' als je je in `/usr/src/linux` bevindt, start een configuratiescript, welke je vele vragen stelt. Het vereist bash, dus verifieer dat bash zich in `/bin/bash`, `/bin/sh`, of `$BASH` bevindt.

Er zijn echter wat plezieriger alternatieven voor '`make config`' en het kan heel goed zijn dat je ze makkelijker en comfortabeler in het gebruik vindt. '`make menuconfig`' is waarschijnlijk de meest gebruikte. Wat je ook kiest, het is het beste als je bekend raakt met de interface, omdat het heel goed mogelijk is dat je eerder terug zult keren, dan je zult denken. Voor degene die "X draaien," kun je '`make xconfig`' proberen, als je Tk hebt geïnstalleerd '`make menuconfig`' is voor degene die (n)curses hebben en de voorkeur zouden geven aan een op tekst gebaseerd menu. Deze interfaces hebben een nogal duidelijk voordeel: Als je het verprutst en tijdens de configuratie een verkeerde keuze maakt, is het heel eenvoudig om terug te gaan en het te corrigeren.

De configuratie-opties zullen met '`make menuconfig`' en `make xconfig`' in hierarchiën verschijnen.

Je bent er klaar voor om de vragen te gaan beantwoorden, gewoonlijk met '`y`' (yes) of '`n`' (no). Device drivers hebben typisch een '`m`' optie. Dit betekent "module," wat inhoudt dat het systeem ze niet direct in de kernel zal compileren, maar als een laadbare module. Een komischer manier om het te beschrijven is als "misschien." Een aantal van de vanzelfsprekende en niet-kritieke opties is hier niet beschreven; zie de sectie "Andere configuratie-opties" voor beknopte beschrijvingen van een paar andere opties. Met '`make menuconfig`', verwissel je met de spatiebalk van selectie.

In 2.0.x en later, is er een '?' optie, die voorziet in een korte beschrijving van de configuratieparameter. Die informatie is waarschijnlijk het meest up-to-date. Hier is een opsomming van een aantal van de belangrijke faciliteiten, in welke hierarchie ze staan, en met een korte beschrijving.

4.3.1 Kernel math emulatie (Processor type and facilities)

Als je geen math processor hebt (je hebt een minimale 386 of 486SX), moet je hier met '`y`' antwoorden. Als je een coprocessor hebt en je zegt dan toch '`y`', maak je er dan niet te veel zorgen om, de coprocessor wordt nog steeds gebruikt en de emulatie wordt genegeerd. Voor iedere moderne computer, zal het antwoord 'no' zijn, maar maak je geen zorgen als je hier per ongeluk 'yes' hebt geantwoord; als het niet nodig is, wordt het niet gebruikt.

4.3.2 Enhanced (MFM/RLL) disk en IDE disk/cdrom support (Block Devices)

Je zult de ondersteuning waarschijnlijk nodig hebben; het betekent dat de kernel standaard PC harddisks, die de meeste mensen hebben, zal ondersteunen. Bij deze driver zijn SCSI-drivers niet inbegrepen; die komen later in de configuratie aan de orde.

Er zal dan worden gevraagd naar de "old disk-only" en "new IDE" drivers. Je zult er hier één van willen kiezen; het belangrijkste verschil is dat de oude driver alleen twee disks op een enkele interface ondersteunt, en de nieuwe een tweede interface en IDE/ATAPI cdromdrives ondersteunt. De nieuwe driver is 4k groter dan de oude en is, naar men mag aannemen, ook "verbeterd," wat betekent dat het afgezien van een verschillend aantal bevattende fouten, het je diskperformance kan verbeteren, vooral als je nieuwere (EIDE-type) hardware hebt.

4.3.3 Netwerkondersteuning (General Setup)

In principe, zou je hier slechts 'y' antwoorden, als je computer zich in een netwerk bevindt, zoals het internet, of je gebruik wilt maken van SLIP, PPP, term, enz. om in te bellen voor toegang tot het internet. Je zou hier echter 'y' moeten antwoorden, want veel packages (zoals het X-windowsysteem) vereisen netwerkondersteuning, ook al is je computer niet op een echt netwerk aangesloten. Later zal je worden gevraagd of je ondersteuning wilt voor een TCP/IP netwerk; antwoord hier nogmaals 'y', als je er niet helemaal zeker van bent.

4.3.4 System V IPC (General Setup)

Één van de beste definities van IPC (Interprocess Communication) staat in de verklarende woordenlijst van het Perl boek. Het is niet verwonderlijk dat een aantal Perl programmeurs het gebruikt om processen met elkaar te laten communiceren, evenals vele andere packages (in 't bijzonder DOOM), dus het is geen goed idee om hier `n` op te antwoorden, tenzij je precies weet wat je aan het doen bent.

4.3.5 Processor familie (Processor type en faciliteiten)

(in oudere kernels: Gebruik de -m486 flag voor 486-specifieke optimalisaties)

Volgens traditie werden hierdoor bepaalde optimalisaties voor een bepaalde processor meegecompileerd; de kernels draaide prima op andere chips, maar de kernel was misschien wat groter. In nieuwere kernels geldt dit echter niet meer, dus je zou op moeten geven voor welke processor je de kernel aan het compileren bent. Een "386" kernel werkt met alle computers.

4.3.6 SCSI-support

Als je SCSI-devices hebt, antwoord je 'y'. Er zal je worden gevraagd naar meer informatie, zoals ondersteuning voor CD-ROM, disks, en wat voor soort SCSI-adapter je hebt. Zie de SCSI-HOWTO voor uitgebreidere details.

4.3.7 Network device support

Als je een netwerkkaart hebt, of je zou SLIP, PPP of een parallelle poortadapter wilen gebruiken om verbinding te maken met het Internet, antwoord je 'y'. Het configuratiescript zal je vragen naar de soort kaart die je hebt, en welk protocol moet worden gebruikt.

4.3.8 Filesystems

Het configureerscript vraagt je vervolgens of je ondersteuning voor de volgende bestandssystemen wilt:

Standard (minix) - Nieuwere distributies maken geen minix bestandssystemen aan, en veel mensen gebruiken het niet, maar het kan nog steeds een goed idee zijn om deze te configureren. Een aantal "rescue disk" programma's maakt er gebruik van, en steeds meer diskettes kunnen een minix bestandssysteem hebben, aangezien het minix bestandssysteem minder moeizaam op een diskette is te gebruiken.

Second extended - Dit is het standaard Linux bestandssysteem. Het is bijna zeker dat je er hier één van hebt, en 'y' moet antwoorden.

msdos - Als je je MS-DOS harddisk partities wilt gebruiken, of MS-DOS geformatteerde diskettes wilt gebruiken, antwoord je 'y'.

Er zijn diverse andere externe soorten besturingssystemen beschikbaar.

/proc - (idee van Bell Labs, denk ik). Men maakt geen proc bestandssysteem op een disk; dit is een bestandssysteeminterface naar de kernel en de processen. Veel programma's die processen weergeven (zoals 'ps') maken er gebruik van. Probeer eens een keer 'cat /proc/meminfo' of 'cat /proc/devices'. Een aantal shells (rc, in het bijzonder) gebruikt voor I/O /proc/self/fd (op andere systemen bekend als /dev/fd). Je zou hier bijna zeker 'y' moeten antwoorden; veel belangrijke linux tools zijn ervan afhankelijk.

NFS - Als je computer op een netwerk is aangesloten, en je wilt bestandssystemen gebruiken die voorkomen op andere systemen met NFS, antwoord dan 'y'.

ISO9660 - Wordt op de meeste CD-ROM's aangetroffen. Als je een CD-ROM drive hebt en je wilt het onder Linux gebruiken, geef je als antwoord 'y'.

Maar ik weet niet welke bestandssystemen ik nodig heb! Ok, typ 'mount'. De uitvoer zal er ongeveer zo uitzien:

```
blah# mount
/dev/hda1 on / type ext2 (defaults)
/dev/hda3 on /usr type ext2 (defaults)
none on /proc type proc (defaults)
/dev/fd0 on /mnt type msdos (defaults)
```

Bekijk iedere regel; het woord naast 'type' is het type bestandssysteem. In dit voorbeeld zijn mijn / en /usr bestandssystemen second extended, Ik gebruik /proc, en er is een diskette gemount door gebruik te maken van het msdos-bestandssysteem.

Je kunt 'cat /proc/filesystems' proberen, als je /proc thans is geactiveerd; je huidige kernel's bestandssystemen zullen worden weergegeven.

De configuratie van zeldzaam gebruikte, niet kritieke bestandssystemen kunnen een bloat kernel veroorzaken; zie de sectie over modules voor een manier om dit te voorkomen en de sectie "Valkuilen" over waarom een bloat kernel niet wenselijk is.

4.3.9 Character devices

Hier activeer je de drivers voor je printer (dat wil zeggen, parallelle printer), busmouse, PS/2 mouse (veel notebooks gebruiken het PS/2 mouse protocol voor hun ingebouwde trackballs), een aantal tapedrives, en andere "character" devices. Antwoord 'y' als dit van toepassing is.

Opmerking: gpm is een programma waarmee het gebruik van de muis buiten het X-windowsysteem voor knippen en plakken tussen virtuele consoles is toegestaan. Het is echt heel aardig als je een seriële muis hebt, omdat het goed naast X kan voortbestaan, maar je hebt voor andere muizen speciale foefjes nodig.

4.3.10 Sound

Als je een groot verlangen voelt om geklap en geblaf te horen, antwoord dan 'y', en je kunt het configuratieprogramma alles over je geluidskaart laten weten. (Een opmerking over de configuratie van een geluidskaart: als je wordt gevraagd of je de volledige versie van de driver wilt installeren, kun je hier 'n' antwoorden en wat kernelgeheugen besparen door alleen de faciliteiten eruit te pikken die je nodig acht).

Als de geluidsondersteuning je menens is, bekijk dan eens de vrij verkrijgbare drivers bij http://www.linux.org.uk/OSS/ en het commerciële Open Sound System bij http://www.opensound.com/.

4.3.11 Andere configuratie-opties

Niet alle configuratie-opties worden hier opgesomd, omdat ze te vaak wijzigen of tamelijk vanzelfsprekend zijn (bijvoorbeeld, 3Com 3C509 ondersteuning voor het compileren van de devicedriver voor deze speciale ethernetkaart). Er bestaat een tamelijk uitgebreide lijst met opties (plus een manier om ze in het `Configure` script te plaatsen) met inzet gestart en beheerd door Axel Boldt (`boldt@math.ucsb.edu`) en het is de online-help. Het is sinds versie 2.0 ook beschikbaar als één groot bestand als de `Documentation/Configure.help` in je Linux kernel source tree.

4.3.12 Kernel hacking

>Vanuit het README bestand van Linus:

de "kernel hacking" configuratie details resulteren meestal in een grotere of langzamere kernel (of beiden), en kan de kernel zelfs minder stabiel maken door een aantal routines te configureren die actief probeert slechte code aan de oppervlakte te laten komen om kernelproblemen op te sporen (kmalloc()). Dus je zou waarschijnlijk 'n' moeten antwoorden op de vragen voor een "production" kernel.

4.4 Wat nu? (De Makefile)

Nadat je de configuratie hebt beëindigd, krijg je een bericht dat je kernel is geconfigureerd en de melding "check the top-level `Makefile` for additional configuration," enz.

Dus bekijk `Makefile`. Je zult het waarschijnlijk niet hoeven wijzigen, maar het kan nooit kwaad het te bekijken. Je kunt de opties ervan ook wijzigen met het commando '`rdev`' zodra de nieuwe kernel op z'n plaats staat. Als het je ontgaat als je het bestand bekijkt, maak je er dan geen zorgen om.

5 Compileren van de kernel

5.1 Opschonen en afhankelijkheden

Als het configuratiescript eindigt, geeft het ook aan een '`make dep`' en (mogelijk) '`clean`' uit te voeren. Dus, doe de '`make dep`'. Dit verzekert je dat alle afhankelijkheden, zoals de include bestanden, op hun plaats staan. Het duurt niet lang, tenzij je computer om te beginnen nogal langzaam is. Je zou voor oudere versies van de kernel een '`make clean` moeten doen, als je klaar bent. Hiermee worden alle object-bestanden en wat andere zaken verwijderd die door een oude versie achter worden gelaten. Vergeet deze stap in ieder geval *niet* voordat je een poging gaat ondernemen om een kernel te hercompileren.

5.2 Compileertijd

Na een `make dep` en een `make clean`, kun je nu een '`make bzImage`' of '`make bzdisk`' opstarten (dit is het onderdeel dat lang duurt). '`make bzImage`' zal de kernel compileren, en (onder andere) een bestand in `arch/i386/boot` met de naam '`bzImage`' achterlaten. Dit is de nieuw gecomprimeerde kernel. '`make bzdisk`' doet hetzelfde, maar het plaatst ook het nieuwe `bzImage` op een diskette, die je hopelijk in drive "A:" deed. '`bzdisk`' is nogal handig voor het testen van nieuwe kernels; als het totaal mislukt (of het gewoon niet goed werkt), verwijder dan gewoon de diskette en boot met je oude kernel. Als je per ongeluk je kernel verwijdert (of iets net zo vreselijks), kan het ook een handige manier zijn om te booten. Je kunt het ook gebruiken om nieuwe systemen te installeren als je de inhoud van de ene naar de andere disk dumpt. ("dit alles en meer! hoeveel zou je NU willen betalen?").

Alle pas halverwege redelijk recente kernels zijn gecomprimeerd, hieruit volgt de '`bz`' aan het begin van de naam. Een gecomprimeerde kernel decomprimeert zichzelf als het wordt uitgevoerd.

In oudere kernels had je de optie niet om een `bzImage` te bouwen; het was gewoon een `zImage`. Die optie is op het moment nog steeds beschikbaar, het is echter, gegeven de grootte van de nieuwere kernels, min of meer verplicht om een `bzImage` te bouwen, omdat de oudere methoden niet met een al te grote kernel om kunnen gaan.

5.3 Andere "make" 's

Met '`make mrproper`' zal een grootser opgezette '`schoonmaak` worden uitgevoerd. Soms is het nodig; je wilt het misschien bij iedere patch doen. Met '`make mrproper`' zal ook je configuratiebestand worden verwijderd, dus misschien wil je een backup maken van (`.config`), als je het als waardevol beschouwd.

Met '`make oldconfig`' zal worden geprobeerd de kernel vanuit een oud configuratiebestand te configureren; het zal het '`make config`' proces voor je doorlopen. Als je nooit eerder een kernel hebt gecompileerd of je hebt geen oud configuratiebestand, dan zou je dit waarschijnlijk niet moeten doen, aangezien je zeer waarschijnlijk je standaardconfiguratie zal willen veranderen.

Zie de sectie over modules voor een beschrijving van '`make modules`'.

5.4 Installeren van de kernel

Nu dat je een kernel hebt die lijkt te werken zoals je het wilt, is het tijd om het te installeren. De meeste mensen gebruiken hiervoor LILO (Linux Loader). '`make bzlilo`' zal de kernel installeren, draai LILO, en je bent er helemaal klaar voor om te booten, MAAR ALLEEN als lilo op de volgende manier op je systeem is geconfigureerd: kernel is `/vmlinuz`, lilo is in `/sbin`, en je lilo config (`/etc/lilo.conf`) stemt daar mee overeen.

Anders moet je LILO op directe wijze gebruiken. Het is een tamelijk makkelijk package om te installeren en om mee te werken, maar het heeft de neiging om mensen met z'n configuratiebestand in de war te brengen. Zoek naar het configuratiebestand (`/etc/lilo/config` voor oudere versies of `/etc/lilo.conf` voor nieuwe versies), en bekijk de huidige setup. Het configuratiebestand ziet er ongeveer zo uit:

```
image = /vmlinuz
    label = Linux
    root = /dev/hda1
    ...
```

De '`image =`' is op de huidige geïnstalleerde kernel ingesteld. De meeste mensen gebruiken `/vmlinuz`. '`label`' wordt door lilo gebruikt om vast te stellen welke kernel of welk besturingssysteem moet worden geboot, en '`root`' is de `/` van dat bepaalde besturingssysteem. Maak een backup van je oude kernel en kopieer het `bzImage` dat je net op z'n plaats hebt gezet (je zou hier '`cp bzImage /vmlinuz`' opgeven als je '`/vmlinuz`' gebruikt). Start lilo dan weer op, op nieuwere systemen kun je gewoon '`lilo`' opstarten, maar op oudere, kan het zijn dat je `/etc/lilo/install` of zelfs `/etc/lilo/lilo -C /etc/lilo/config` op moet geven.

Als je meer zou willen weten over de configuratie van LILO, of je hebt LILO niet, haal dan de nieuwste versie vanaf je favoriete ftp-site en volg de instructies op.

Om één van je oude kernels vanaf de harddisk te booten (een andere manier om jezelf te redden voor 't geval je de nieuwe kernel hebt verprutst), kopieer je de regels hieronder (en voeg ze in): '`image = xxx`' in het configuratiebestand van LILO onderaan het bestand, waarbij je '`image = xxx`' wijzigt in '`image = yyy`';hierbij staat '`yyy`' voor de volledige padnaam van het bestand, waarin je je backup van de kernel bewaarde.

Wijzig dan de regel '`label = zzz`' in '`label = linux-backup`' en herstart `lilo`. Misschien moet je nog een regel in het configuratiebestand plaatsen zoals '`delay=x`', waar x staat voor het aantal tienden seconden, dat LILO voor het booten moet wachten, zodat je het kunt onderbreken. (bijvoorbeeld met de shift-toets), en het label in kunt tikken van de backup boot-image (voor 't geval er onplezierige dingen gebeuren).

6 Patchen van de kernel

6.1 Toepassen van een patch

Periodieke upgrades van de kernel worden als patches gedistribueerd. Als je bijvoorbeeld versie 1.1.45 hebt, en je merkt dat er een '`patch46.gz`' voor is, betekent het dat je naar versie 1.1.46 kunt upgraden door toepassing van de patch. Misschien wil je eerst een backup van de sourcetree maken ('`make clean`' en vervolgens '`cd /usr/src; tar zcvf old-tree.tar.gz linux`' zal een gecomprimeerd tar-archief voor je maken).

Dus, verdergaand met het voorbeeld van hierboven, laten we ervan uitgaan dat je '`patch46.gz`' in `/usr/src` hebt. `cd` naar `/usr/src` en doe een '`zcat patch46.gz | patch -p0`' (of '`patch -p0 < patch46`' als de patch niet is gecomprimeerd). Je zult van alles voorbij zien vliegen (of fladderen als je systeem zo langzaam is) om je te laten weten dat het aan het proberen is om brokken toe te passen en of het daarin slaagt of niet. Meestal gaan deze acties te snel voorbij om ze te kunnen lezen, en ben je er niet helemaal zeker van of het wel of niet werkte, dus misschien wil je de `-s` flag aan `patch` opgeven, welke `patch` aangeeft alleen de foutmeldingen te rapporteren (je krijgt niet zoveel als bij het "hé, mijn computer doet eindelijk eens wat voor de verandering!" gevoel, maar het kan zijn dat je hier de voorkeur aan geeft ..). Om die onderdelen te bekijken die misschien niet zo soepel zijn verlopen, cd je naar `/usr/src/linux` en zoek je naar bestanden met een `.rej` extensie. Een aantal versies van `patch` (oudere versies welke kunnen zijn gecompileerd met een inferieur bestandssysteem) laat de verwerpingen met een `#` extensie achter. Je kunt '`find`' gebruiken om ze voor je op te sporen;

```
find .  -name '*.rej' -print
```

drukt alle bestanden in de huidige directory of elke subdirectory met de `.rej` extensie naar standaarduitvoer af.

Als alles goed ging, doe je een '`make clean`', '`config`', en '`dep`' zoals in sectie 3 en 4 werd beschreven.

Er zijn heel wat opties voor het `patch` commando. Zoals hierboven genoemd, zal `patch -s` alle berichten behalve de foutmeldingen onderdrukken. Als je je kernelsource op een andere plaats dan `/usr/src/linux` bewaart, zal er met `patch -p1` (in die directory) een zuivere patch worden uitgevoerd. Andere `patch` opties zijn goed gedocumenteerd in de manual page.

6.2 Als er iets fout gaat

(Opmerking: deze sectie refereert voornamelijk naar nogal oude kernels)

Het meest voorkomende probleem dat zich voordeed was wanneer een patch een bestand met de naam '`config.in`' wijzigde en het er niet helemaal goed uitzag, omdat je de opties wijzigde om aan je computer aan te passen. Dit is verholpen, maar het kan zijn dat je het met een oudere release nog aan zult treffen. Bekijk het bestand `config.in.rej` om het te herstellen, en zie wat er van de originele patch over is gebleven. Kenmerkend is dat de wijzigingen aan het begin van de regel met een '`+`' en een '`-`' worden gemarkeerd. Kijk naar de regels die erdoor worden omsloten, en denk eraan terug of ze met '`y`' of '`n`' werden ingesteld. Wijzig nu je `config.in`, en verander daar waar van toepassing de '`y`' in een '`n`' en de '`n`' in een '`y`' Tik in

```
patch -p0 < config.in.rej
```

en als het rapporteert dat het succesvol was (zonder gebreken), dan kun je verdergaan met de configuratie en compilatie. Het bestand `config.in.rej` blijft behouden, maar je kunt het verwijderen.

Als je verdere problemen tegenkomt, kan het zijn dat je een patch in de verkeerde volgorde hebt geïnstalleerd. Als patch de melding '`previously applied patch detected: Assume -R?`' geeft, ben je waarschijnlijk aan het proberen om een patch toe te passen welke lager is dan het huidige versienummer; als je '`y`' antwoordt, zal het proberen je source te degraderen, en zal hier waarschijnlijk niet in slagen; dus je zult een volledige nieuwe versie van de source tree nodig hebben (wat in eerste instantie niet eens zo'n slecht idee zou zijn geweest).

Om een patch achteraf te verwijderen, gebruik je '`patch -R`' op de originele patch (het toepassen ongedaan maken).

Als patches echt verkeerd blijken te zijn, kun je het beste opnieuw beginnen met een nog onaangetaste source tree (bijvoorbeeld vanuit één van de `linux-x.y.z.tar.gz` bestanden).

6.3 De .orig bestanden zien kwijt te raken

Na slechts een paar patches, zullen de `.orig` bestanden zich beginnen op te stapelen. Een 1.1.51 tree die ik bijvoorbeeld ooit had, was voor het laatst bij 1.1.48 opgeschoond. Het verwijderen van de .orig bestanden bespaarde me meer dan een halve meg.

```
find .  -name '*.orig' -exec rm -f {} ';'
```

zal dit voor je regelen. Versies van `patch` die `#` gebruiken voor verwerpingen, maken gebruik van een tilde in plaats van `.orig`.

Er zijn betere manieren om af te geraken van de `.orig` bestanden, die afhankelijk zijn van GNU `xargs`:

```
find .  -name '*.orig' | xargs rm
```

of de "heel veilige maar een beetje uitgebreidere" methode:

```
find . -name '*.orig' -print0 | xargs --null rm --
```

6.4 Andere patches

Er zijn andere patches (Ik zal ze "nietstandaard" noemen) dan die Linus distribueert. Als je deze toepast, kan het zijn dat de patches van Linux niet correct werken en dan zul je ze alsnog moeten verwijderen, de source of de patch moeten herstellen, een nieuwe sourcetree moeten installeren, of een combinatie van het bovenstaande. Dit kan erg frustrerend worden, dus als je de source niet wilt wijzigen (met de kans op een zeer slechte uitkomst), verwijder dan de niet-standaard patches voordat je die van Linux toepast, of installeer gewoon een nieuwe tree. Vervolgens kun je zien of de niet-standaard patches nog steeds werken. Als dat niet zo is, zit je vast aan een oude kernel, het spelen met de patch of de source om het aan het werk te krijgen, of zul je moeten wachten (of bedelen) tot er een nieuwe versie van de patch uitkomt.

Hoe algemeen zijn de patches die zich niet in de standaarddistributie bevinden? Je zult waarschijnlijk van ze horen. Ik was gewend de noblink patch voor mijn virtuele consoles te gebruiken, omdat ik een hekel heb aan knipperende cursors. (Deze patch wordt (of tenminste werd) vaak bijgewerkt voor nieuwe kernelreleases). Van de meeste nieuwe devicedrivers, die als laadbare modules worden ontwikkeld, is de frequentie van "niet-standaard"patches echter aanmerkelijk aan het afnemen.

7 Aanvullende packages

Je linuxkernel heeft vele faciliteiten die niet in de kernelsource zelf zijn uitgelegd; deze faciliteiten worden kenmerkend door externe packages gebruikt. Een aantal van de meest gebruikelijke faciliteiten wordt hier opgesomd.

7.1 kbd

De linux console heeft waarschijnlijk meer faciliteiten dan het toekomt. Hiertussen bevindt zich de mogelijkheid om van lettertypen te verwisselen, je toetsenbord opnieuw in te delen, tussen video-modes te schakelen (in nieuwere kernels), enz. Het kbd package bestaat uit programma's die de gebruiker in staat stellen om dit allemaal te doen, plus nog vele lettertypen en toetsenbordindelingen voor bijna ieder toetsenbord, en het is vanaf dezelfde sites beschikbaar waar de kernelsource te vinden is.

7.2 util-linux

Rik Faith (`faith@cs.unc.edu`) heeft een grote verzameling linux utility's bijeen gebracht, door een eigenaardig toeval, met de naam util-linux. Deze worden nu door Andries Brouwer (`util-linux@math.uio.no`) beheerd. Beschikbaar via anonieme ftp vanaf sunsite.unc.edu in `/pub/Linux/system/misc`, het bevat programma's zoals `setterm`, `rdev`, en `ctrlaltdel`, die relevant zijn voor de kernel. Zoals Rik zegt, *installeer het niet zonder erbij na te denken ;* je hoeft niet alles dat zich in het package bevindt, te installeren, en het zou heel goed ernstige problemen kunnen veroorzaken als je dit wel doet.

7.3 hdparm

Zoals met vele packages, was dit ooit een kernelpatch en ondersteunde programma's. De patches haalde het tot in de officiële kernel, en de programma's voor het optimaliseren en spelen met je harddisk worden afzonderlijk gedistribueerd.

7.4 gpm

gpm staat voor 'general purpose mouse.' Dit programma staat je toe om tussen virtuele consoles tekst te knippen en te plakken en wat andere dingen te doen met een grote diversiteit aan muistypes.

8 Een aantal valkuilen

8.1 make clean

Als je nieuwe kernel echt vreemde dingen doet na een routine kernel-upgrade, bestaat de kans dan je vergat `make clean` uit te voeren voordat je de nieuwe kernel compileerde. De symptomen kunnen van alles zijn, van je systeem dat ineens crasht, vreemde I/O problemen, te lage performance. Wees er zeker van dat je ook een `make dep` doet.

8.2 Zeer grote of langzame kernels

Als je kernel een boel geheugen opslurpt, te groot is, en/of het compileren eeuwig duurt, ook al heb je een nieuwe Quadbazillium-III/4400 die eraan werkt, dan heb je waarschijnlijk erg veel onnodig spul (device drivers, bestandssystemen, enz) geconfigureerd. Als je het niet gebruikt, configureer het dan niet, want het neemt geheugen in beslag. Het meest opvallende van kernel bload is het extreme in en uitswappen van geheugen naar de disk; als je disk een heleboel lawaai maakt en het niet één van die oude Fujitsu Eagles is, die klinken alsof er een straalvliegtuig landt als je je computer uitzet, kijk dan nog eens naar je kernelconfiguratie.

Je kunt erachter komen hoeveel geheugen je kernel gebruikt door de totale hoeveelheid geheugen in je machine af te trekken van de hoeveeelheid van "total mem" in `/proc/meminfo` of de uitvoer van het commando '`free`'.

8.3 De parallelle poort werkt niet/mijn printer werkt niet

Configuratie-opties voor PC's zijn: Selecteer als eerste, onder de categorie 'General Setup', 'Parallel port support' en 'PC-style hardware'. Selecteer dan onder 'Character devices', 'Parallel printer support'.

En dan zijn er nog de namen. Linux 2.2 noemt de printerdevices anders dan in voorgaande releases. Het komt hierop neer dat, als je onder je oude kernel een `lp1` had, het onder je nieuwe kernel waarschijnlijk een `lp0` is. Gebruik '`dmesg`' of doorzoek de logs in `/var/log` om erachter te komen.

8.4 Kernel compileert niet

Als het niet compileert, dan is het waarschijnlijk dat er een patch mislukte, of je source is op énén of andere manier verknoeid. Het kan ook zijn dat je niet de juiste versie van gcc hebt, of deze kan ook verknoeid zijn (de include bestanden kunnen bijvoorbeeld fout zijn). Zorg ervoor dat de symbolische links, die Linux in de `README` beschrijft, juist zijn ingesteld. In het algemeen geldt, dat als een standaardkernel niet compileert, er iets ernstig mis is met het systeem, en opnieuw installeren van bepaalde tools is waarschijnlijk noodzakelijk.

In een aantal gevallen kan gcc door hardwareproblemen crashen. De foutmelding zal iets zijn als "xxx exited with signal 15" en het zal er gewoonlijk zeer mysterieus uitzien. Ik zou dit waarschijnlijk niet ter sprake hebben gebracht, behalve dat het me een keer overkwam - Ik had wat slecht cache geheugen, en de compiler kon nu en dan willekeurig weigeren. Probeer als eerste gcc opnieuw te installeren als je problemen ervaart. Je zou alleen achterdochtig moeten zijn, als je kernel goed compileert met externe cache uitgezet, een verminderde hoeveelheid RAM, enz.

Het schijnt mensen te storen wanneer er wordt gesuggereerd, dat er problemen met hun hardware zijn. Ik verzin dit niet. Er is een FAQ voor – het is te vinden bij `http://www.bitwizard.nl/sig11/`.

8.5 De nieuwe kernelversie lijkt niet te booten

Je hebt LILO niet gedraaid of het is niet juist geconfigureerd. Één ding dat me eens "overkwam", was een probleem in het configuratiebestand; het gaf aan '`boot = /dev/hda1`' in plaats van '`boot = /dev/hda`' (Dit kan in het begin echt hinderlijk zijn, maar zodra je een werkend configuratiebestand hebt, zou je het niet meer hoeven te wijzigen).

8.6 Je vergat LILO te draaien, of je systeem boot helemaal niet

Oeps! Het beste wat je hier kunt doen is met een diskette of CDROM te booten en een andere opstartbare diskette aan te maken (zoals '`make zdisk`' zou doen). Je zult moeten weten waar je root (/) bestandssysteem zich bevindt en van welk type het is (b.v. second extended, minix). In het voorbeeld hieronder zul je ook moeten weten op welk bestandssysteem je `/usr/src/linux` source-tree zich bevindt, het type, en waar het normaal gesproken wordt gemount.

In het volgende voorbeeld, is `/` `/dev/hda1`, en het bestandssysteem met `/usr/src/linux` is `/dev/hda3`, normaal gesproken gemount onder `/usr`. Het zijn allebei second extended bestandssystemen. De werkende kernel image in `/usr/src/linux/arch/i386/boot` wordt `bzImage` genoemd.

De bedoeling is, dat als er een functionerend `bzImage` is, het mogelijk is om dat voor de nieuwe diskette te gebruiken. Een ander alternatief, welke wel of niet beter kan werken, (dit hangt af van de speciale methode waarin je je systeem hebt verknoeid) wordt na het voorbeeld besproken.

Boot om te beginnen vanaf een boot/root diskset of rescuedisk, en mount het bestandssysteem waarin zich de werkende kernel-image bevindt:

```
mkdir /mnt
mount -t ext2 /dev/hda3 /mnt
```

Als `mkdir` je de melding geeft dat de directory al bestaat, negeer het dan gewoon. `cd` nu naar de plaats waar de werkende kernel-image stond. Merk op dat

```
/mnt + /usr/src/linux/arch/i386/boot - /usr = /mnt/src/linux/arch/i386/boot
```

Plaats een geformatteerde disk in drive “A:” (niet je boot- of rootdisk!), dump de image naar de disk, en configureer het voor je root bestandssysteem:

```
cd /mnt/src/linux/arch/i386/boot
dd if=bzImage of=/dev/fd0
rdev /dev/fd0 /dev/hda1
```

`cd` naar / en unmount het normale `/usr` bestandssysteem:

```
cd /
umount /mnt
```

Je zou je systeem nu zoals gewoonlijk vanaf deze diskette op moeten kunnen starten. Vergeet na het opstarten, lilo niet te draaien (of wat je ook verkeerd deed)!

Zoals hierboven genoemd, is er nog een ander alternatief. Als je een werkende kernel-image in `/` hebt, (bijvoorbeeld `/vmlinuz`), kun je dat voor een bootdisk gebruiken. Uitgaande van alle bovenstaande condities, en dat mijn kernel-image `/vmlinuz` is, maak je gewoon deze wijzigingen aan, in het voorbeeld hierboven: verander `/dev/hda3` in `/dev/hda1` (het `/` bestandssysteem), `/mnt/src/linux` in `/mnt`, en `if=bzImage` in `if=vmlinuz`. De opmerking die uitleg geeft hoe `/mnt/src/linux` kan worden afgeleid, kan worden genegeerd.

LILO met grote drives gebruiken, (meer dan 1024 cylinders) kan problemen veroorzaken. Zie de LILO mini-HOWTO of documentatie hierover voor hulp.

8.7 Het geeft de melding ‘warning: bdflush not running’

Dit kan een ernstig probleem zijn. Beginnend met kernelrelease na 1.0 (rond 20 Apr 1994), werd een programma met de naam ‘`update`’ bijgewerkt, welke periodiek de buffers van het bestandssysteem opschoont. Haal de sources van ‘`bdflush`’ op (je zou het moeten kunnen vinden daar waar je je kernelsource vandaan hebt gehaald), en installeer het (je wilt je systeem waarschijnlijk onder de oude kernel draaien als je hiermee bezig bent). Het installeert zichzelf als ‘`update`’ en na een reboot, zou de nieuwe kernel er niet langer problemen mee moeten hebben.

8.8 Ik krijg mijn IDE/ATAPI CD-ROM drive niet aan de praat

Vreemd genoeg krijgen een heleboel mensen hun ATAPI drives niet werkend, waarschijnlijk omdat er een aantal dingen verkeerd kan gaan.

Als je CD-ROM drive het enige apparaat op een bepaalde IDE interface is, moet het als “master” of “slave” zijn gejumperd. Dit is vermoedelijk de meest voorkomende fout.

Creative Labs heeft nu IDE-interfaces op hun geluidskaarten gezet. Dit leidt echter tot het interessante probleem dat terwijl een aantal mensen slechts één interface heeft , hebben velen twee IDE-interfaces op hun moederborden ingebouwd (meestal op IRQ15), dus het is een algemene gewoonte om van de soundblaster interface een derde IDE poort te maken (IRQ11, is me verteld).

Dit veroorzaakt problemen met linux gezien versies 1.2.x geen derde IDE-interface ondersteunen (er is ondersteuning te beginnen ergens in 1.3.x series maar dat is development, denk daaraan, en het doet geen auto-probe). Om dit te omzeilen, heb je een paar keuzes.

Als je al een tweede IDE-poort hebt, bestaat de kans dat je het niet gebruikt, of er zich nog geen twee devices op bevinden. Haal de ATAPI-drive van de geluidskaart af en bevestig het aan de tweede interface. Je kunt de interface van de geluidskaart vervolgens de-activeren, wat je hoe dan ook een IRQ bespaart.

Als je geen tweede interface hebt, jumper de interface van de geluidskaart (niet het geluidsdeel van de geluidskaart) dan als IRQ15, de tweede interface. Het zou moeten werken.

8.9 Het geeft rare meldingen over ‘obsolete routing requests’

Haal nieuwe versies op van het `route` programma en enige andere programma’s die er zijn voor route manipulatie. `/usr/include/linux/route.h` (dat eigenlijk een bestand in `/usr/src/linux` is), is gewijzigd.

8.10 Firewall werkt niet in 1.2.0

Upgrade naar tenminste versie 1.2.1.

8.11 “Not a compressed kernel Image file”

Gebruik het bestand `vmlinux` die je in `/usr/src/linux` hebt aangemaakt, niet als je boot-image; het juiste bestand is `[..]/arch/i386/boot/bzImage`.

8.12 Problemen met console terminal na het upgraden tot 1.3.x

Wijzig het woord `dumb` in `linux` in de console termcap entry in `/etc/termcap`. Het kan ook zijn dat je een terminfo entry moet maken.

8.13 Het lijkt erop dat ik dingen na de kernel-upgrade niet kan compileren

De linux kernelsource bevat een aantal include bestanden (die met een `.h` eindigen), waarnaar wordt verwezen door de standaard include bestanden in `/usr/include`. Er wordt als volgt naar verwezen (waar `xyzzy.h` iets in `/usr/include/linux` zou zijn):

```
#include <linux/xyzzy.h>
```

Normaal gesproken, is er een link met de naam `linux` in `/usr/include` naar de directory `include/linux` van je kernelsource (`/usr/src/linux/include/linux` op het systeem). Als deze link niet voorkomt, of naar de verkeerde plaats verwijst, zal het meeste helemaal niet worden gecompileerd. Als je besloot de kernelsource te verwijderen, omdat het te veel ruimte op de disk in beslag nam, zal dit uiteraard een probleem zijn. Een andere manier waarop het fout kan gaan is door bestandspermissies; als je `root` een umask heeft, die andere gebruikers niet toestaat, standaard de bestanden te zien, en je pakte de kernelsource uit met de `p` (preserve filemodes) optie, dan zullen die gebruikers ook niet in staat zijn om de C compiler te gebruiken. Alhoewel je het `chmod` commando zou kunnen gebruiken om dit te herstellen, is het waarschijnlijk makkelijker om de include bestanden opnieuw uit te pakken. Je kunt dit op dezelfde manier doen zoals je in het begin met de hele source deed, alleen met een aanvullend argument:

```
blah# tar zxvpf linux.x.y.z.tar.gz linux/include
```

Opmerking: "`make config`" zal de `/usr/src/linux` opnieuw aanmaken als het er niet is.

8.14 Verhogen van limieten

De volgende paar *voorbeeld*commando's kunnen handig zijn voor degenen die zich afvragen hoe ze bepaalde software-limieten kunnen verhogen die door de kernel worden opgelegd:

```
echo 4096 > /proc/sys/kernel/file-max
echo 12288 > /proc/sys/kernel/inode-max
echo 300 400 500 > /proc/sys/vm/freepages
```

9 Opmerking voor upgrade naar versie 2.0.x, 2.2.x

Kernel versies 2.0.x en 2.2.x introduceerde heel wat wijzigingen voor de kernel-installatie. Het bestand `Documentation/Changes` in de 2.0.x source tree bevat informatie waarmee je bekend zou moeten zijn als je naar één van deze versies gaat upgraden. Je zult zeer waarschijnlijk verscheidene packages moeten upgraden, zoals gcc, libc, en SysVInit, en misschien een aantal systeembestanden moeten wijzigen, dus wees hier op voorbereid. Echter, geen paniek.

10 Modules

Laadbare kernelmodules kunnen geheugen besparen en de configuratie vergemakkelijken. De strekking van modules is gegroeid dat het bestandssystemen, ethernetkaartdrivers, tapedrivers, printer drivers, en meer bevat.

10.1 Installeren van de module-utility's

De module utility's zijn beschikbaar vanwaar je je kernelsource vandaan hebt gehaald als `modutils-x.y.z.tar.gz`; kies het hoogste patchlevel `x.y.z` dat gelijk is of lager dan dat van je huidige kernel. Pak het uit met '`tar zxvf modutils-x.y.z.tar.gz`', `cd` naar de directory, het maakt (`modutils-x.y.z`) aan, bekijk de `README`, en voer de installatie-instructies ervan uit (die meestal erg eenvoudig zijn, zoals `make install`). Je zou nu de programma's `insmod`, `rmmod`, `ksyms`, `lsmod`, `genksyms`, `modprobe`, en `depmod` in `/sbin` moeten hebben. Test de utility's met de "hw" voorbeelddriver in `insmod`, als je dat wilt; bekijk het bestand `INSTALL` in die subdirectory voor details.

`insmod` voegt een module in, in de draaiende kernel. Modules hebben meestal een `.o` extensie; de voorbeelddriver die hierboven werd genoemd, heeft de naam `drv_hello.o`, dus om dit in te voegen, zou men intikken '`insmod drv_hello.o`'. Om te modules te zien die de kernel op dit moment gebruikt, gebruik je `lsmod`. De uitvoer ziet er ongeveer zo uit:

```
blah# lsmod
Module:         #pages:  Used by:
drv_hello          1
```

'`drv_hello`' is de naam van de module, het gebruikt één pagina (4k) van het geheugen, en er zijn op dit moment geen andere kernelmodules afhankelijk van. Om deze module te verwijderen, gebruik je '`rmmod drv_hello`'. Merk op dat `rmmod` de *naam van een module* verlangt, geen bestandsnaam; je krijgt dit van de uitvoer van `lsmod`. De strekking van de andere utility's voor de modules staan in hun manual pages gedocumenteerd.

10.2 Met de kernel gedistribueerde modules

Sinds versie 2.0.30, is bijna alles als een laadbare module beschikbaar. Zorg er eerst voor dat je ze niet in de reguliere kernel configureert, om ze te gebruiken; dat wil zeggen, beantwoord 't niet met `y` tijdens '`make config`'. Compileer een nieuwe kernel en reboot ermee. `cd` dan nogmaals naar `/usr/src/linux`, en doe een '`make modules`'. Hiermee worden alle modules gecompileerd die je niet in de kernelconfiguratie hebt gespecificeerd, en links ernaar in `/usr/src/linux/modules` geplaatst. Je kunt ze direct vanuit die directory gebruiken of '`make modules_install`' uitvoeren, waarmee ze in `/lib/modules/x.y.z` worden geïnstalleerd, hierbij staat `x.y.z` voor de kernelrelease.

Dit kan vooral handig zijn met bestandssystemen. Het kan zijn dat je het minix of msdos bestandssysteem niet vaak gebruikt. Als ik bijvoorbeeld een msdos (huiver) diskette trof, zou ik `insmod /usr/src/linux/modules/msdos.o`, en dan `rmmod msdos` als ik klaar ben. Deze procedure bespaart gedurende de normale bewerking ongeveer 50k RAM in de kernel. Een kleine aantekening voor het minix bestandssysteem: je zou het *altijd* direct in de kernel moeten configureren, voor gebruik in "rescue" disks.

11 Tips en trucs

11.1 Doorsturen van uitvoer van de make of patch commando's

Als je logs van wat die '`make`' of '`patch`' commando's deden, zou willen hebben, kun je de uitvoer naar een bestand doorsturen. Zoek als eerste uit onder welke shell je draait: '`grep root /etc/passwd`' en zoek naar iets als '`/bin/csh`'.

Als je sh of bash gebruikt, zal

```
(commando) 2>&1 | tee (uitvoerbestand)
```

een kopie van de uitvoer van `(commando)` in het bestand '`(uitvoerbestand)`' plaatsen.

Gebruik voor csh of tcsh,

```
(command) |& tee (uitvoerbestand)
```

Voor rc (Opmerking: je gebruikt rc waarschijnlijk niet) is het

```
(command) >[2=1] | tee (uitvoerbestand)
```

11.2 Voorwaardelijke kernel-installatie

Naast het gebruiken van diskettes, zijn er nog verscheidene methoden om een nieuwe kernel uit te testen zonder de oude kernel aan te roeren. In tegenstelling tot andere Unix-soorten, heeft LILO de mogelijkheid om een kernel vanaf iedere plaats op de disk te booten. (als je een grote disk (500 MB of meer) hebt, lees dan alsjeblieft de LILO-documentatie door over hoe dit problemen kan veroorzaken). Dus, als je iets vergelijkbaars als het volgende aan het einde van je LILO configuratiebestand toevoegt,

```
image = /usr/src/linux/arch/i386/boot/bzImage
    label = new_kernel
```

kun je ervoor kiezen een nieuwe gecompileerde kernel te draaien zonder dat je je oude `/vmlinuz` aanroert (na het draaien van `lilo`, natuurlijk). De makkelijkste manier om LILO te vertellen dat het een nieuwe kernel moet booten is om de linker shift-toets tijdens het opstarten in te drukken, (als je `LILO` op het scherm ziet staan, en niets anders) waardoor je een prompt krijgt. Op dit punt kun je '`nieuwe_kernel`' opgeven om de nieuwe kernel te booten.

Als je verscheidene verschillende kernel source-trees tegelijkertijd op je systeem wilt behouden (dit kan *heel veel* diskruimte innemen; wees voorzichtig), is de meest gebruikelijke manier om ze `/usr/src/linux-x.y.z` te noemen, waar `x.y.z` voor de kernelversie staat. Je kunt een sourcetree dan met een symbolische link "selecteren"; bijvoorbeeld, '`ln -sf linux-1.2.2 /usr/src/linux`' zou de 1.2.2 tree de huidige maken. Zorg ervoor dat het laatste argument van `ln` geen echte directory is (oude symbolische links zijn prima), voordat je op deze manier een symbolische link maakt; het resultaat zal je namelijk anders niet hetgene geven wat je ervan zou verwachten.

11.3 Kernel updates

Russell Nelson (`nelson@crynwr.com`) vat de wijzigingen in nieuwe kernelreleases samen. Ze zijn beknopt, en je zou er misschien voor een upgrade naar willen kijken. Ze zijn beschikbaar via anonieme ftp vanaf `ftp.emlist.com` in `pub/kchanges` of via de URL

```
http://www.crynwr.com/kchanges
```

12 Andere relevante HOWTO's die nuttig kunnen zijn

- Sound-HOWTO: geluidskaarten en utility's
- SCSI-HOWTO: alles over SCSI-controllers en -devices
- NET-2-HOWTO: netwerk
- PPP-HOWTO: PPP-netwerken in het bijzonder
- PCMCIA-HOWTO: over de drivers voor je notebook
- ELF-HOWTO: ELF: wat het is, het converteren..
- Hardware-HOWTO: overzicht van ondersteunde hardware
- Module mini-HOWTO: meer over kernelmodules
- Kerneld mini-HOWTO: over kerneld
- BogoMips mini-HOWTO: voor 't geval je 't je af mocht vragen

13 Diversen

13.1 Auteur

De auteur en beheerder van de Linux Kernel-HOWTO is Brian Ward (`bri@cs.uchicago.edu`). Stuur me alsjeblieft je opmerkingen, aanvullingen, correcties (Correcties zijn voor mij in het bijzonder het belangrijkst).

Je kunt een kijkje nemen op mijn 'home page' op één van deze URL's:

```
http://www.math.psu.edu/bri/
http://blah.math.tu-graz.ac.at/~bri/
```

Ook al probeer ik zo attent mogelijk te zijn met mail, denk er alsjeblieft aan dat ik er iedere dag *heel veel* van krijg, dus het kan even duren voor ik je antwoord. Vooral als je me mailt met een vraag, probeer dan alsjeblieft extra je best te doen in je bericht duidelijk en gedetailleerd te zijn. Als je me schrijft over niet werkende hardware (of iets dergelijks) moet ik weten wat je hardwareconfiguratie is. Als je een fout rapporteert, zeg dan niet slechts "Ik heb dit geprobeerd, maar het gaf een foutmelding;" Ik moet weten wat de foutmelding was. Ik zou ook willen weten welke versie van de kernel, gcc, en libc je gebruikt. Als je me slechts vertelt dat je één of andere versie van een bepaalde distributie gebruikt, zal me dat niet veel zeggen. Het maakt me niet uit als je eenvoudige vragen stelt; denk eraan, als je niets vraagt, kun je ook nooit een antwoord krijgen! Ik zou graag iedereen willen bedanken die me feedback heeft gegeven.

Als je vraag niets met de kernel te maken heeft, of in een taal staat die ik niet begrijp, kan het zijn dat ik niet antwoord.

Als je me een mail hebt gestuurd en binnen redelijke tijd (drie weken of meer) geen antwoord hebt gekregen, dan bestaat de kans dat ik je bericht per ongeluk heb verwijderd of iets van die strekking (sorry). Probeer het alsjeblieft opnieuw.

Ik krijg erg veel mail over zaken die eigenlijk te maken hebben met hardware of hardwareproblemen. Dat is prima, maar probeer er alsjeblieft aan te denken dat ik niet bekend ben met alle in de wereld te krijgen hardware. Ik gebruik AMD processors, Adaptec en Sybios SCSI-controllers, en IBM SCSI-disks.

Versie -0.1 werd geschreven op 3 oktober 1994. Dit document is beschikbaar in SGML, PostScript, TeX, roff, en plain-text formaten.

13.2 Te doen

De "Tips en trucs" sectie is wat klein. Ik hoop het uit te kunnen breiden door suggesties van anderen. Zo ook "Aanvullende packages." Meer debugging/crash herstel info nodig.

13.3 Contributies

Een klein deel van Linus' README (kernelhacking opties) is inbegrepen. (Bedankt, Linus!)

`uc@brian.lunetix.de` (Ulrich Callmeier): patch -s en xargs.

`quinlan@yggdrasil.com` (Daniel Quinlan): correcties en aanvullingen in vele secties.

`nat@nat@nataa.fr.eu.org` (Nat Makarevitch): mrproper, tar -p, vele andere zaken

`boldt@math.ucsb.edu` (Axel Boldt): verzamelde beschrijvingen van kernelconfiguratie-opties op het net; voorzag me vervolgens in een lijst

`lembark@wrkhors.psyber.com` (Steve Lembark): multiple boot suggestie

`kbriggs@earwax.pd.uwa.edu.au` (Keith Briggs): een aantal correcties en suggesties `rmcguire@freenet.columbus.oh.us` (Ryan McGuire): makeables aanvullingen

`dumas@excalibur.ibp.fr` (Eric Dumas): Franse vertaling

`simazaki@ab11.yamanashi.ac.jp` (Yasutada Shimazaki): Japanse vertaling

`jjamor@lml.ls.fi.upm.es` (Juan Jose Amor Iglesias): Spaanse vertaling

`mva@sbbs.se` (Martin Wahlen): Zweedse vertaling

`jzp1218@stud.u-szeged.hu` (Zoltan Vamosi): Hongaarse vertaling

`bart@mat.uni.torun.pl` (Bartosz Maruszewski): Poolse vertaling

`donahue@tiber.nist.gov` (Michael J Donahue): typos, winnaar van de “sliced bread competition”

`rms@gnu.ai.mit.edu` (Richard Stallman): “vrije” documentatie concept/distributie notitie

`dak@Pool.Informatik.RWTH-Aachen.DE` (David Kastrup): iets over NFS

`esr@snark.thyrsus.com` (Eric Raymond): diverse juweeltjes

De mensen die me mail zonden met vragen en problemen waren ook zeer behulpzaam.

13.4 Copyright-melding, Licentie en al dat soort zaken

Copyright © Brian Ward, 1994-1999.

Het is toegestaan kopieën van deze handleiding te distribueren, op voorwaarde dat de copyright-melding en deze permissie-melding op alle kopieën behouden blijft.

Het is toegestaan gewijzigde versies van deze handleiding te kopieëren en te distribueren onder de condities voor letterlijk kopieëren, op voorwaarde dat het afgeleide werk onder de voorwaarden van een permissie-melding identiek is aan deze vermelding. Vertalingen vallen onder de categorie “gewijzigde versies.”

Garantie: Geen.

Aanbevelingen: Commerciële herdistributie is toegestaan en wordt aangemoedigd; het wordt echter sterk aangeraden dat de herdistributeur contact opneemt met de auteur voor de herdistributie, in het belang van zaken up-to-date te houden (je zou me een kopie van hetgeen je aan het maken bent toe kunnen sturen terwijl je er mee bezig bent). Vertalers worden ook geadviseerd om contact op te nemen met de auteur voor het vertalen. De afgedrukte versie ziet er mooier uit. Recycle.

14 Andere formaten van dit document

Deze sectie is geschreven door *Al Dev* (op de site `<http://www.aldev.8m.com>` mirrors bij `<http://aldev.webjump.com>`, *angelfire* `<http://www.angelfire.com/country/aldev0>`, *geocities* `<http://www.geocities.com/alavoor/index.html>`, *virtualave* `<http://aldev0.virtualave.net>`, *50megs* `<http://aldev0.50megs.com>`, *theglobe* `<http://members.theglobe.com/aldev1/index.html>`, *NBCi* `<http://members.nbci.com/alavoor>`, *Terrashare* `<http://aldev.terrashare.com>`, *Fortunecity* `<http://members.fortunecity.com/aldev>`, *Freewebsites* `<http://aldev.freewebsites.com>`, *Tripod* `<http://members.tripod.lycos.com/aldev>`, *Spree* `<http://members.spree.com/technology/aldev>`, *Escalix* `<http://www.escalix.com/freepage/aldev>`, *Httpcity* `<http://www.httpcity.com/aldev/index.html>`, *Freeservers* `<http://aldev.freeservers.com>`)

Dit document is in 12 verschillende formaten geschreven, te weten - DVI, Postscript, Latex, Adobe Acrobat PDF, LyX, GNU-info, HTML, RTF(Rich Text Format), Plain-text, Unix man pages, een enkel HTML bestand en SGML.

- Je kunt dit HOWTO document als een enkele tarball ophalen in HTML, DVI, Postscript of SGML formaten vanaf - <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/> en <http://www.linuxdoc.org/docs.html##howto>
- Het gewone tekstformaat is te vinden in: <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO> en <http://www.linuxdoc.org/docs.html##howto>
- Het enkele HTML bestandsformaat is te vinden in: <http://www.linuxdoc.org/docs.html##howto>
- Vertalingen naar andere talen zoals Frans, Duits, Spaans, Chinees, Japanees zijn te vinden in <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO> en <http://www.linuxdoc.org/docs.html##howto> Elke hulp om naar andere talen te vertalen is welkom.

Het document is geschreven met een tool genaamd "SGML-Tools"wat kan worden opgehaald vanaf <http://www.sgmltools.org> Voor het compileren van de source gebruik je de volgende opdrachten

- sgml2html Kernel-HOWTO.sgml (voor het genereren van het html bestand)
- sgml2rtf Kernel-HOWTO.sgml (voor het genereren van het RTF bestand)
- sgml2latex Kernel-HOWTO.sgml (voor het genereren van het latex bestand)

LaTeX documenten kunnen naar PDF bestand worden geconverteerd door eenvoudigweg Postscript uitvoer te produceren met behulp van **sgml2latex** (en dvips) en de uitvoer door als volgt de Acrobat **distill** (<http://www.adobe.com>) opdracht uit te voeren:

```
bash$ man sgml2latex
bash$ sgml2latex filename.sgml
bash$ man dvips
bash$ dvips -o filename.ps filename.dvi
bash$ distill filename.ps
bash$ man ghostscript
bash$ man ps2pdf
bash$ ps2pdf input.ps output.pdf
bash$ acroread output.pdf &
```

Of je kunt gebruik maken van de Ghostscript opdracht **ps2pdf**. ps2pdf is een work-alike voor bijna alle functionaliteit van Adobe's Acrobat Distiller product: het converteert PostScript bestanden naar Portable Document Format (PDF) bestanden. **ps2pdf** is geïmplementeerd als een zeer klein opdrachtscript (batch file) dat Ghostscript aanroept, waarbij het een speciaal öutput device"genaamd **pdfwrite** selecteert. Voor gebruik van ps2pdf moest het pdfwrite device zijn opgenomen in de makefile wanneer Ghostscript werd gecompileerd; zie de documentatie over het bouwen van Ghostscript voor details.

Dit howto document is te vinden op:

- <http://sunsite.unc.edu/LDP/HOWTO/Kernel-HOWTO.html>

Dit document is ook te vinden op de volgende mirrorsites:

- <http://www.caldera.com/LDP/HOWTO/Kernel-HOWTO.html>
- <http://www.WGS.com/LDP/HOWTO/Kernel-HOWTO.html>
- <http://www.cc.gatech.edu/linux/LDP/HOWTO/Kernel-HOWTO.html>

- <http://www.redhat.com/linux-info/ldp/HOWTO/Kernel-HOWTO.html>
- Andere mirrorsites in je buurt (netwerk-adres-wijsheid) zijn te vinden op <http://sunsite.unc.edu/LDP/hmirrors.html> selecteer een site en ga naar de directory /LDP/HOWTO/Kernel-HOWTO.html

Om het document te bekijken in dvi formaat, gebruik je het programma xdvi. Het programma xdvi is onder RedHat Linux te vinden in het tetex-xdvi*.rpm package welke kan worden gelokaliseerd via het menu ControlPanel | Applications | Publishing | TeX . Geef de volgende opdracht om het dvi document te lezen

```
xdvi -geometry 80x90 howto.dvi
man xdvi
```

En vergroot/verklein het venster met de muis. Gebruik om te navigeren de Pijltjestoetsen, Page Up, Page Down toetsen, tevens kun je gebruik maken van de lettertoetsen 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' voor respectievelijk naar boven, naar beneden, centreren, volgende pagina, vorige pagina, enz. Druk de 'x' toets in om het expert menu uit te zetten.

Met het programma 'gv' (ghostview) of 'ghostscript' kun je het postscript bestand lezen. Onder RedHat is het ghostscript programma te vinden in het ghostscript*rpm package en het programma gv in het gv*rpm package wat te vinden is via het menu ControlPanel | Applications | Graphics. Het programma gv is veel gebruikersvriendelijker dan ghostscript. Ghostscript en gv zijn ook beschikbaar voor andere platformen, zoals OS/2, Windows 95 en NT. Je kunt dit document zelfs onder deze platformen bekijken.

- Ghostscript voor Windows 95, OS/2, en voor alle andere besturingssystemen is op te halen vanaf <http://www.cs.wisc.edu/~ghost>

Geef de opdracht

```
gv howto.ps
ghostscript howto.ps
```

om het postscript document te kunnen lezen.

Met Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web browser of een van de andere 10 webbrowsers kun je het document in HTML formaat lezen.

De uitvoer van LyX, de latex uitvoer, kun je lezen met LyX, een X-Window frontend naar latex.