Linux Kernel Modules Installatie HOWTO

rhw@bigfoot.com

Vertaald door: Ellen Bokhorst

Beschrijft de installatie van Linux kernelmodules.

Inhoudsopgave

1. Doel van dit document

Mijn ervaring met Linux en modules bestaat hieruit dat de bestaande documenten er niet in slagen mij naar tevredenheid een uitleg te geven over hoe men Linux met modules geconfigureerd en met succes werkend instelt. De procedure uitgelegd in dit document is verscheidene malen met succes toegepast op zowel mijn eigen systeem als over het Internet waarbij ik iemand aanwijzigingen gaf om een bepaalde feature proberen werkend te krijgen waarvoor een driver nodig was die alleen in de vorm van een module werd geleverd.

Op mijn eigen systeem draait een RedHat 4.1 distibutie van Linux, en ik ontwikkelde de procedure vanuit deze setup. Ik heb het sindsdien met succes op systemen geïnstalleerd waarop diverse Slackware distributies draaide, en op een systeem met een Debian distributie. De benodigde procedure om modules correct onder Linux te configureren op al deze drie is hierin gedocumenteerd.

Ik heb dezelfde procedure recent met RedHat 4.2 toegepast, maar met inconsistente resultaten op schijnbaar identieke systemen. Ik heb nog niet vastgesteld wat het probleem is, dus kan in dit stadium geen garanties geven of het wel of niet op je systeem zal werken.

2. Veronderstellingen

- Voor de stappen in dit document kunnen worden toegepast, moet de lezer een werkende Linux installatie tot zijn beschikking hebben waarbij men als gebruiker root bij de Linux prompt kan komen, aangezien de meeste te nemen stappen alleen kunnen worden ondernomen door de genoemde gebruiker.
- De bestaande kernel kan zo worden gecompileerd dat er wel of geen gebruik wordt gemaakt van modules, en er kunnen zelfs foutmeldingen worden getoond tijdens de opstartprocedure als resultaat van geconfigureerde modules die op dat moment niet beschikbaar zijn, op voorwaarde dat aan de bovenstaande conditie is voldaan.
- Er wordt in dit document verondersteld dat het startpunt van de bronstructuur voor de huidige kernel is te vinden in `/usr/src/linux` en er wordt tevens vanuit gegaan dat dit ook het startpunt is van een uit te voeren reeks opdrachten.

3. Compileertijd versnellen

Als je machine met 16 of meer Megabyte RAM is uitgerust, dan kan er een aardige snelheidswinst worden geboekt door de kernel twee of meer modules parallel te laten compileren. Dit zal de belasting op de machine verhogen onderwijl de kernel wordt gecompileerd, maar het zal de tijd bekorten die de compilatie in beslag zal nemen.

Voordat je deze methode kunt gebruiken, moet je de hoeveelheid aanwezig RAM op je machine controleren, want als je dit te hoog instelt, zal het de compilatie vertragen. Ervaring heeft aangetoond dat de optimale waarde afhankelijk is van de hoeveelheid RAM in je systeem overeenkomstig de volgende formule, tenminste voor systemen tot 32 Megabyte RAM, alhoewel dit wat conservatief kan zijn voor systemen met grotere hoeveelheden RAM:

N = [RAM in Megabytes] / 8 + 1

Voor degenen die niet zo van wiskunde houden, staan de waarden voor de gebruikelijke hoeveelheden RAM in de volgende tabel:

Tabel 1. Voorbeeldtabel

RAM Size	Te gebruiken waarde
16 Meg	3
24 Meg	4
32 Meg	5
40 Meg	6
48 Meg	7
56 Meg	8
64 Meg	9
80 Meg	11
96 Meg	13
112 Megs	15
128 Megs	17

Wanneer je hebt besloten tot het juiste aantal, wijzig je het bestand `/usr/src/linux/Makefile` en zoek je naar de regel waarin thans staat:

```
MAKE=make
```

Vervang het door:

```
MAKE=make -j N
```

waar `N` het eerder vastgestelde getal is.

4. De kernel hercompileren voor gebruik met modules

De kernel kan opnieuw worden geconfigureerd voor gebruik met modules voor alles behalve het bestandssysteem dat als root wordt gemount (in de meeste gevallen is dit het ext2 bestandssysteem).

Er zijn echter bepaalde items die moeilijk juist in te stellen zijn als modules, dus raad ik het volgende aan om in de kernel te compileren:

- Ethernet hardware drivers.
- SCSI CD-ROM drivers.

Aan de andere kant bestaan er bepaalde drivercombinaites die ALLEEN als modules functioneren, vooral combinaties bestaande uit één of meer uit de volgende groep:

- Een Parallel Printer,
- Een Parallel Poort drive, zoals de IOMEGA ZipDrive of JazzDrive, of de BackPack CD-ROM drive, en
- De PLIP Daemon.

Je zult moeten besluiten wat je in de kernel compileert en wat als modules, maar neem bovenstaande punten daarbij in overweging. De werkelijke keuzes worden gemaakt tijdens de compilatie, door de tweede instructie uit de volgende reeks:

```
cd /usr/src/linux
make menuconfig
make dep clean modules modules_install zImage
```

Wanneer dat is gebeurd, moeten de module afhankelijkheden in kaart worden gebracht. Dit wordt bewerkstelligd met de volgende opdracht:

```
depmod -a
```

De nieuwe kernel moet nu in de bootreeks worden ingevoegd. Ik veronderstel dat de lezer voor dit doel gebruik maakt van LILO, aangezien dit de enige bootloader is waarmee ik ervaring heb.

Ik raad je aan de nieuw gecompileerde kernel NIET automatisch als de standaard Linux kernel in te voegen want als het niet functioneert, het dan zeer moeilijk is de Linux setup te herstellen zonder een volledig nieuwe installatie uit te voeren, wat niet is aan te bevelen. Daarom heb ik het volgende record in het bestand `/etc/lilo.conf`:

```
image=/usr/src/linux/arch/i386/boot/zImage
  label=new
  alias=n
  read-only
  vga=ask
  optional
```

Dit record geeft aan dat er een OPTIONELE bootoptie is (welke zal worden genegeerd als de image in kwestie niet bestaat) wat het bestand `/boot/newlinux` zal booten als dit wordt gekozen, en het biedt de mogelijkheid de videomodus te selecteren waarin het zal worden geboot.

Veronderstel het bestaan van het bovenstaande record in `/etc/lilo.conf`, de gereviseerde kernel bevindt zich na de compilatie reeds op de juiste lokatie en het kan via de volgende opdracht worden geïnstalleerd:

```
lilo
```

Nadat dat is gebeurd, moet de lezer als volgt de volgende stappen nemen die relevant zijn met betrekking tot de gekozen distributie:

5.1. Debian of RedHat voor gebruik van modules configureren

Voordat deze stappen worden uitgevoerd, wordt verondersteld dat die in "De kernel hercompileren voor gebruik met modules" reeds zijn opgevolgd.

De Debian en RedHat distributies hebben identieke bootprocedures, dus hebben ze ook identieke procedures voor het configureren van modules.

1. Ingelogd als root, gebruik je je favoriete teksteditor om een nieuw bestand met de naam `/etc/rc.d/init.d/modules.init` aan te maken met de volgende inhoud:

   ```
   # Module initialisatie.
   #
   # Start de module auto-loading daemon.
   /sbin/kerneld

   # Mount alle thans niet gemounte auto-mounted partities.
   /sbin/mount -a
   ```

2. Na het aanmaken van het bovenstaande bestand, voer je de volgende stappen uit onderwijl ingelogd als root:

   ```
   cd /etc/rc.d
   chmod 755 init.d/*
   cd rc3.d
   ln -s ../init.d/modules.init 05modules.init
   ```

Het systeem kan nu opnieuw worden opgestart, en wanneer het dit doet, zal het constateren dat modules volledig zijn geïmplementeerd.

5.2. Slackware voor gebruik van modules configureren

Voordat deze stappen worden uitgevoerd, wordt verondersteld dat die in "De kernel hercompileren voor gebruik met modules" reeds zijn opgevolgd.

Het bestand `/etc/rc.d/rc.M` moet als volgt worden gewijzigd:

1. Zo ongeveer op regel 18, staat:

   ```
   # Blank screen after 15 minutes idle.
   /bin/setterm -blank 15
   ```

Voeg direct daarna, de volgende paragraaf in, beginnend en eindigend met de gebruikelijke lege regels:

```
# Laad de kernel module auto-loader.
/sbin/kerneld
```

2. Ongeveer 12 regels verder staat het volgende:

   ```
   # if there is no /etc/HOSTNAME, fall back on this default:
   ```

   Onmiddelijk hiervoor, voeg je de volgende paragraaf in, wederom met de gebruikelijk lege regels rondom:

   ```
   # Mount de resterende niet gemounte auto-mount drives.
   /sbin/mount -a
   ```

Sla het bestand op wanneer de wijzigingen zijn aangebracht.

Voor Slackware zijn geen verdere aanpassingen nodig.

5.3. Andere distributies voor gebruik van modules configureren

Voordat deze stappen worden uitgevoerd, wordt verondersteld dat die in “De kernel hercompileren voor gebruik met modules” reeds zijn opgevolgd.

De precieze procedure voor andere distributies zijn we niet te weten gekomen, maar waarschijnlijk is dit gelijk aan het voorgaande. Om vast te stellen welke, bekijk je als volgt een weergave van de inhoud van de directory `/etc/rc.d`:

```
cd /etc/rc.d
ls -l *.d rc.*
```

Uit de resulterende weergave, kun je een van de volgende drie opties selecteren:

1. Als in deze weergave een directory met de naam `init.d` voorkomt en nog een aantal direectory’s met namen die overeenkomen met `rc?.d` waar het vraagteken kan worden vervangen door enkele cijfers, en waarin GEEN bestand is opgenomen met de naam `rc.M`, dan kan die distributie worden geconfigureerd voor modules door de procedure te volgen onder de titel “Debian of RedHat voor gebruik van modules configureren”.
2. Als in deze weergave geen directory is opgenomen met de naam `init.d`, maar wel een bestand met de naam `rc.M` dan kan die distributie voor modules worden geconfigureerd door de procedures te volgen die terug te vinden zijn onder de titel “ Slackware voor gebruik van modules configureren”.
3. Als deze weergave met geen van bovenstaande criteria overeenkomt, dan komt in deze distributie een bootscript voor die niet in deze HowTo wordt behandeld. In dat geval wordt je uitgenodigd contact op te nemen met de auteur van dit document om advies.