Separate Kernel Trees

Frodo Looijaard, *frodol@dds.nl*
Vertaald door: Ellen Bokhorst, *bokkie@nl.linux.org*
v1.1, 12 juli 1998

Dit document beschrijft een setup waarmee de installatie en het gebruik van verscheidene aparte kernel-structuren mogelijk is, zelfs als ze hetzelfde releasenummer hebben.

1 Introductie

Dit document zou van nut voor je kunnen zijn als je aan één of andere ontwikkeling gerelateerd aan de kernel werkt, waarbij je een nieuwe compiler versie uitprobeert of pre-versies installeert.

Een aantal distributies kunnen reeds iets vergelijkbaars als met de hieronder setup hebben. Als de directory-structuren in `/lib/modules` zijn genaamd 2.0.34 voor kernelversies 2.0.34, is dit waarschijnlijk niet het geval. Als ze 2.0.34-iets zijn genoemd, zal je dit document waarschijnlijk niet nodig hebben.

Ik heb dit alleen met kernel 2.0.x getest. Ik zou het waarderen te horen of het tevens met kernels 2.1.x werkt.

2 Setup

We zullen naar kernelversies refereren als een aaneenschakeling van releasenummer en compilatienummer, gescheiden door een koppelteken om het gebruik van verscheidene kernels met hetzelfde releasenummer mogelijk te maken.

Voorbeeld: Kernelversie 2.0.32, derde compilatie, zou worden genoemd 2.0.32-3

Tijdens de uitvoering van `/etc/rc` (bij het opstarten van het systeem), moeten verscheidene stappen worden genomen om er zeker van te zijn dat dit zal werken (VERSION is een versienummer zoals hierboven is uitgelegd):

- `/usr/src/linux moet worden gelinkt naar /usr/src/VERSION`
- `/lib/modules/current moet worden gelinkt naar /lib/modules/VERSION`

Je kunt VERSION met het volgende stukje code vaststellen:

```
kernel_version ()
{
  local IFS_safe="$IFS"
  IFS="#-$IFS"
  set -- `uname -v`
  IFS="$IFS_safe"
  echo `uname -r`-$2
}
```

Door gebruik te maken van deze functie, kun je iets als volgt aan je rc-bestanden toevoegen (deze code is wat paranoïde, maar beter op veilig spelen dan spijt achteraf)

```
echo "Selecteren van de huidige modules..."
KERNELD=yes
```

```
VERSION=`kernel_version`
if [ -d "/lib/modules/$VERSION" ] ; then
  if [ -L /lib/modules/current ] ; then
    rm -f /lib/modules/current
  fi
  if [ ! -e /lib/modules/current ] ; then
    ln -s "$VERSION" /lib/modules/current
  else
    echo "FOUT: /lib/modules/current bestaat en is geen symbolische link."
    echo "      Het laden van een module kan onverwachte resultaten opleveren, misschien"
    echo "      zelfs in gegevensverlies resulteren. We zullen kerneld niet laden. Verwacht"
    echo "      hieronder veel fouten."
    KERNELD=no
  fi
else
  echo "FOUT:  /lib/modules/$VERSION niet gevonden."
  echo "       Kerneld kan niet worden gestart, modules zullen niet"
  echo "       automatisch worden geladen"
  echo "       Verwacht hierna veel foutmeldingen."
  KERNELD=no
fi


echo "Selecteren van de huidige kernel..."
VERSION=`kernel_version`
if [ -d /usr/src/linux-"$VERSION" ] ; then
  if [ -L /usr/src/linux ] ; then
    rm -f /usr/src/linux
  fi
  if [ ! -e /usr/src/linux ] ; then
    ln -s linux-"$VERSION" /usr/src/linux
  else
    echo "WAARSCHUWING: /usr/src/linux kan naar een andere kernel dan die waarvan we bootte"
    echo "              zijn gelinkt. Dit geeft mogelijk wat kleine verrassingen, maar"
    echo "              zou niet echt een probleem moeten zijn."
  fi
else
  echo "WARNING: /usr/src/linux-$VERSION niet gevonden. /usr/src/linux is wellicht niet"
  echo "         naar de huidige kernel gelinkt. Dit geeft mogelijk wat kleine verrassingen"
  echo "         maar zou niet echt een probleem moeten zijn."
fi
```

Bovendien moet het bestand `/etc/conf.modules` aanwezig zijn en op z'n minst de volgende informatie bevatten (soms wordt dit bestand `modules.conf` genoemd):

Waar het dependency bestand kan worden gevonden depfile=/lib/modules/current/modules.dep

Dit zijn de paths naar de modules specifiek voor de kernelversie

- `path[fs]=/lib/modules/current/fs`
- `path[misc]=/lib/modules/current/misc`
- `path[net]=/lib/modules/current/net`
- `path[scsi]=/lib/modules/current/scsi`

- `path[block]=/lib/modules/current/block`
- `path[ipv4]=/lib/modules/current/ipv4`

Dit zijn de paths naar aanvullende modules die kernel onafhankelijk zijn:

- `path[fs]=/lib/modules/extra/fs`
- `path[misc]=/lib/modules/extra/misc`
- `path[net]=/lib/modules/extra/net`
- `path[scsi]=/lib/modules/extra/scsi`
- `path[block]=/lib/modules/extra/block`
- `path[ipv4]=/lib/modules/extra/ipv4`

Dit zijn de paths naar modules die tijdens het booten voor alle kernelversies worden ingevoegd. Door het maken van symbolische links naar ../current/... kun je nog steeds verschillende modules voor verschillende kernels laden.

- `path[boot]=/lib/modules/boot`

Dit zijn de paths naar modules die tijdens het booten voor specifieke kernels worden ingevoegd.

- `path[boot]=/lib/modules/current/boot`

Om er zeker van te zijn dat dit schema werkt, moet je voorzichtig zijn bij het compileren van een nieuwe kernel:

- Verzeker je ervan dat de combinatie van het release en compilatienummer uniek is. Je kunt het compilatienummer instellen door het nummer_min_één in <Kernel-base>/.version in te voeren (verwijder dit bestand om nummer 1 te gebruiken).
- Plaats iedere kernel-structuur in de directory `/usr/src/VERSION`. Wees voorzichting wanneer je een nieuwe kernel-structuur vanuit een tarball uitpakt, aangezien het wordt uitgepakt in de directory linux en mogelijk een oude kernel-structuur overschrijft! Over het algemeen werkt dit voor mij:

```
# Overtuig jezelf ervan dat niemand iets aan het compileren is!
cd /usr/src
rm linux
tar zxfv /tmp/linux-2.0.34.tar.gz
mv linux linux-2.0.34-1
ln -s linux-2.0.34-1 linux
echo 0 > linux/.version
# echo er altijd één minder dan het compileernummer aan .version!
```

Of, als de structuur reeds beschikbaar is:

```
cd /usr/src
cp -a linux-2.0.34-1 linux-2.0.34-2
ln -sn linux-2.0.34-2 linux
cd linux
make mrproper
# in .version zou het juiste nummer moeten staan na de laatste compilatie; het is beter
# hier zeker van te zijn!
echo 1 > .version
```

Na het aanroepen van module_install, hernoem je `/lib/modules/RELEASE` in `/lib/modules/VERSION` (RELEASE is hierbij het kernel release nummer), of gebruik je de hieronderstaande kleine patch:

```
*** /tmp/linux/Makefile Wed Feb  4 19:41:45 1998
--- linux/Makefile      Wed Feb  4 20:04:45 1998
***************
*** 275,279 ****
  modules_install:
        @( \
!       MODLIB=/lib/modules/$(VERSION).$(PATCHLEVEL).$(SUBLEVEL); \
        cd modules; \
        MODULES=""; \
--- 275,279 ----
  modules_install:
        @( \
!       MODLIB=/lib/modules/$(VERSION).$(PATCHLEVEL).$(SUBLEVEL)-`cat .version`; \
        cd modules; \
        MODULES=""; \
```

3 Copyright en disclaimer