RedHat Linux KickStart HOWTO

Martin Hamilton <martinh@gnu.org>, Vertaald door: Ellen Bokhorst <bokkie@nl.linux.org> v0.2, 11 januari 1999

In deze HOWTO wordt beknopt beschreven hoe het RedHat Linux *KickStart* systeem te gebruiken om snel grote aantallen identieke Linux boxen te installeren. Voor geavanceerde gebruikers beschrijven we hoe de Kickstart installatieprocedure naar eigen inzicht aan kan worden gepast en geven we een beknopte leidraad voor het zelf samenstellen van RPM packages.

Inhoudsopgave

1 Copyright

2 Homepage

Als je aan dit document via een Linux HOWTO mirrorsite of een CD-ROM bent gekomen, wil je wellicht de *KickStart HOWTO home page* <http://wwwcache.ja.net/dev/kickstart/> er nog op nazien of hier een nieuwere versie van in omloop is.

3 Introductie

RedHat Linux versie 5 komt met een weinig bekend (en tot nu, niet erg gedocumenteerde) faciliteit genaamd *KickStart*. Hiermee kun je (bijna) alles van de RedHat Linux installatie automatiseren, waaronder:

- Selecteren van de taal
- Netwerkconfiguratie en selecteren van de distributiebron
- Toetsenbordkeuze
- Bootloader installatie (b.v. lilo)
- Diskpartitionering en aanmaak bestandssysteem
- Muiskeuze
- X Window systeem serverconfiguratie
- Keuze van de tijdszone
- Keuze van een (initieel) root wachtwoord
- Welke packages te installeren

Scherpzinnige RedHat gebruikers zullen zich inmiddels waarschijnlijk hebben gerealiseerd dat dit in belangrijke mate de hoofdstappen zijn die een handmatige installatie van een RedHat Linux systeem inhouden. KickStart maakt het je mogelijk een script te schrijven voor het reguliere installatieproces, door de informatie die je normaal gesproken op het toetsenbord op zou geven in een configuratiebestand te plaatsen.

Maar wacht - er is meer :-)

Na het normale installatieproces te hebben beëindigd, laat KickStart je ook een lijst specificeren met opdrachten op niveau van de shell die je zou willen hebben uitgevoerd. Dit betekent dat je automatisch extra lokale software kunt installeren die niet als onderdeel van RedHat Linux wordt gedistribueerd (ja, er zijn zelfs meer vrij verkrijgbare softwareprogramma's dan datgene dat je bij de RedHat distributie krijgt. Een aantal kunnen vanwege wettelijke gronden niet door RedHat worden gedistribueerd, b.v. de `ssh` en `PGP` encryptie systemen). Mogelijk heb je die nodig voor een volledig functioneel besturingssysteem.

4 Eerste vereisten

Er zijn twee benaderingen bij een KickStart installatie - één daarvan is eenvoudigweg je KickStart configuratiebestand naar een RedHat bootdiskette kopiëren. De andere is gebruik te maken van een reguliere bootdiskette en je Kickstart configbestand via het netwerk ophalen.

In beiden gevallen heb je nodig:

1. Intel (i386) klasse computers - waarmee KickStart alleen schijnt te werken tijdens moment van schrijven.
2. KickStart configbestand - hier zullen we het in de volgende sectie over hebben.
3. RedHat bootdisk - bij voorkeur uit de *updates* directory, om voordeel te hebben van eventuele correcties/updates van drivers.
4. DNS records voor de IP-adressen die je zal gaan gebruiken - optioneel, maar zal de installatie onderbreken doordat het je vraagt naar de domeinnaam van je computer.

Als je je configbestand over het netwerk op wilt halen, zal je het via NFS moeten exporteren, dit is de enige toegangsmethode die op het moment wordt ondersteund. Het configbestand laat je een andere NFS server specificeren waar het de RedHat distributie zelf van ophaalt.

Je kunt voor je computer een statisch IP-adres configureren - b.v. een speciaal adres gereserveerd voor KickStart installaties. Als alternatief kun je KickStart opgeven een BOOTP/DHCP server te gebruiken om het IP-adres op te halen als je het niet als vaststaand adres in het configbestand wilt coderen. Een aantal servers zal nieuwe adressen in een gegeven bereik automatisch toewijzen, zoals b.v. *de CMU BOOTP server met dynamische adresseringsextensies* `<ftp://ftp.ntplx.net/pub/networking/bootp>`.

Meer informatie over NFS en BOOTP/DHCP is te vinden in Appendix A.

5 Instellen van een bootdiskette

In essentie is alles wat je moet doen je KickStart configbestand naar */ks.cfg* op de RedHat bootdiskette kopiëren, b.v.

```
mcopy ks.cfg a:
```

De RedHat bootdiskette is echter tamelijk vol, en wellicht bemerk je dat je een aantal van de andere bestanden moet verwijderen om ruimte te maken voor het KickStart configbestand. Het lukte mij voldoende ruimte vrij te maken door de diverse berichtbestanden te verwijderen die normaal gesproken worden weergegeven door de `SYSLINUX` bootloader, b.v.

```
mdel a:\*.msg
```

Een andere benadering zou kunnen zijn de drivers voor een deel van de hardware die je toch niet hebt weg te gooien, zie de sectie over het aanpassen van de bootdiskette hieronder.

Wellicht dat je ook *syslinux.cfg*, het `SYSLINUX` configbestand wilt wijzigen. Ook deze is op de RedHat bootdiskette in de top level directory terug te vinden. De volgende *syslinux.cfg* bijvoorbeeld zal er voor zorgen dat automatisch, zonder de normale onderbreking, de KickStart mode wordt ingegaan als de computer opstart:

```
default ks
prompt 0
label ks
  kernel vmlinuz
  append ks=floppy initrd=initrd.img
```

Naar alle waarschijnlijkheid wil je je boot- en aanvullende diskettes baseren op de meest recente diskimages die beschikbaar zijn in *updates/i386* op je lokale RedHat mirrorsite. Oudere images kunnen fouten bevatten of bieden minder ondersteuning voor hardwaredrivers.

6 Het KickStart configbestand

Het configbestand bestaat uit drie hoofdsecties:

1. Systeem info, b.v. diskpartitionering en netwerkconfig
2. Te installeren RedHat packages
3. Post-installatie uit te voeren shellopdrachten

Er zijn nog een aantal andere mogelijkheden waar we het hier niet over zullen hebben, maar die **wellicht** wel werken. Kijk voor meer informatie in het voorbeeld KickStart config in *misc/src/install/ks.samp* en *doc/README.ks* onder de top level *i386* RedHat distributie directory op je CD-ROM of lokale RedHat mirror site.

6.1 Systeeminfo

De beschikbare directives die ik heb gebruikt zijn:

lang

Taalconfiguratie, b.v. voor Engels

```
lang en
```

network

Netwerkconfiguratie, b.v. gebruik van BOOTP/DHCP

```
network --bootp
```

nfs

NFS server en directory van waaruit te installeren, b.v.

```
nfs --server chicken.swedish-chef.org /mnt/cdrom
```

om de NFS server *chicken.swedish-chef.org* te gebruiken en de RedHat distributie vanuit de directory */mnt/cdrom* proberen te mounten.

keyboard

Selecteer type toetsenbord, b.v. voor UK toetsenborden

```
keyboard uk
```

zerombr

Wis de Master Boot Record - verwijdert een eventueel bestaande bootloader voor besturingssystemen op je disk

clearpart

Wis bestaande partities - b.v. om alle bestaande diskpartities voor de installatie te verwijderen.

```
clearpart --all
```

part

Partitioneer de disk, b.v. voor het aanmaken van een rootbestandssysteem van 500MB

```
part / --size 500
```

install

Voor een nieuwe installatie van RedHat Linux.

mouse

Stel de te gebruiken muis in, b.v. voor een PS/2 of compatibele "bus mouse"

```
mouse ps/2
```

timezone

Stel de tijdszone in, b.v. voor de lokale tijd in de UK

```
timezone --utc Europe/London
```

rootpw

Instelling van het initiële rootwachtwoord, afgeleid van een eerder opgegeven versleuteld wachtwoord

```
rootpw --iscrypted XaacoeGPmf/A.
```

lilo

Installeer de LILO bootloader, b.v. in de Master Boot Record

```
lilo --location mbr
```

/%packages

Te installeren packages - zie hieronder.

%post

Post-installatie shellopdrachten - zie hieronder.

De directory waar KickStart zoekt naar de RedHat distributie moet een subdirectory *RedHat* bevatten, die de RedHat distributiestructuur voor het platform in kwestie bevat. In het voorbeeld hiervoor, zouden we bijvoorbeeld de volgende bestanden en directory's te zien moeten krijgen:

```
/mnt/cdrom/RedHat
/mnt/cdrom/RedHat/base
/mnt/cdrom/RedHat/contents
/mnt/cdrom/RedHat/i386
/mnt/cdrom/RedHat/instimage
/mnt/cdrom/RedHat/RPMS
/mnt/cdrom/RPM-PGP-KEY
```

Als je vanaf een CD-ROM installeert in plaats van vanaf het netwerk, dan ziet de inhoud er ongeveer zo uit:

```
RedHat
RedHat/base
RedHat/contents
RedHat/i386
RedHat/instimage
RedHat/RPMS
RPM-PGP-KEY
```

Als je de RedHat distributie hebt voor meerdere architecturen (b.v. op een NFS server - zijn ze te groot om meer dan één versie van een architectuur op een enkele CD-ROM te passen), zal je bemerken dat iedere distributie dezelfde bestanden en directory's onder een subdirectory heeft, b.v.

```
alpha/RPM-PGP-KEY
i386/RPM-PGP-KEY
sparc/RPM-PGP-KEY
```

Er zou een bestand `architecture`/Redhat/`architecture` moeten zijn, b.v. *i386/Redhat/i386*.

Met Perl is het heel eenvoudig als je je eigen versleutelde wachtwoorden wilt aanmaken, b.v.

```
% perl -e 'print crypt("schmurrdegurr", "Xa") . "\n";'p
```

Andere opties (of niet ter sprake gebrachte opties) die ik niet heb geprobeerd:

cdrom

Installeer vanaf CD-ROM in plaats van vanaf het netwerk.

device

Expliciete declaratie van details, b.v.

```
device ethernet 3c509 --opts "io=0x330, irq=7"
```

Alternatieve waarden van `device` zijn onder andere `scsi` voor SCSI controllers en `cdrom` voor eigen CD-ROM stations.

upgrade

Upgrade een bestaande installatie in plaats van het uitvoeren van een nieuwe installatie.

xconfig

Configureer de X Window server, grafische kaart en monitor.

b.v.

```
xconfig --server "Mach64" --monitor "tatung cm14uhe"
```

Ik ben niet zo diep op die laatste ingegaan omdat ik niet van plan ben X op de console te draaien of op één van mijn met KickStart geconfigureerde computers. Er is me verteld dat de uitvoering van `xconfig` binnen KickStart zelf wat zwak is, maar dezelfde functionaliteit is ook vanaf de opdrachtregel via `Xconfigurator` beschikbaar - dus wellicht dat je dit het beste over kunt laten aan het post-installatie script.

Wanneer we alle stukjes en beetjes bijelkaar plaatsen ziet het KickStart config bestand er ongeveer zo uit:

```
lang en
network --static --ip 198.168.254.253 --netmask 255.255.255.0
  --gateway 198.168.254.1 --nameserver 198.168.254.2
nfs --server chicken.swedish-chef.org /mnt/cdrom
keyboard uk
zerombr yes
clearpart --all
part / --size 500
part swap --size 120
install
mouse ps/2
timezone --utc Europe/London
rootpw --iscrypted XaacoeGPmf/A.
lilo --location mbr
```

Een deel van de RedHat documentatie refereert naar een aanroep van de `network` directive welke in de praktijk niet echt werkt: `network -option`. De juiste aanroep is `network` gevolgd door `-static`, `-bootp` of `-dhcp`. De BOOTP en DHCP opties zijn verschillend - - in de mate dat ze zelfs van andere code gebruik maken.

Je kunt de `-grow` parameter toevoegen aan een `part` directive om aan te geven dat het OK is dat de partitie verder toeneemt dan de grootte die je opgeeft. Dit heeft waarschijnlijk slechts zin als één partitie met `-grow` is gemarkeerd.

6.2 Te installeren packages

Het begin van de sectie packages van het KickStart configuratiebestand wordt aangegeven door de aanwezigheid van een `%packages` directive op een aparte regel. Dit wordt gevolgd door één of beiden typen packages specifiers - individuele packages kunnen worden geïnstalleerd door de naam van de RPM ervan op te geven (exclusief de informatie betreffende de versie en het platform), en groepen packages kunnen worden geïnstalleerd door de naam van de groep waartoe ze behoren op te geven.

Hier is een voorbeeld `packages` sectie voor een KickStart configbestand:

```
%packages
@ Base
netkit-base
bind-utils
ncftp
rdate
tcp_wrappers
traceroute
cmu-snmp
```

Wat zijn deze groepen? Er zijn een aantal groepen standaard gedefinieerd in een bestand genaamd *base/comps* onder de top level directory van de RedHat distributie. Dit zijn degenen die tijdens dit schrijven van toepassing waren:

- Base
- Printer Support
- X Window System
- Mail/WWW/News Tools
- DOS/Windows Connectivity
- File Managers
- Graphics Manipulation
- X Games
- Console Games
- X multimedia support
- Console Multimedia
- Print Server
- Networked Workstation
- Dialup Workstation
- News Server
- NFS Server
- SMB (Samba) Connectivity
- IPX/Netware(tm) Connectivity
- Anonymous FTP/Gopher Server
- Web Server
- DNS Name Server
- Postgres (SQL) Server
- Network Management Workstation
- TeX Document Formatting
- Emacs
- Emacs with X windows
- C Development
- Development Libraries
- C++ Development
- X Development

- Extra Documentation

Het zal je opvallen dat ze met de diverse configuraties corresponderen waar je gedurende een handmatige installatie om wordt gevraagd. Een aantal packages in een gegeven packagegroep komen ook in andere groepen voor. Je kunt meerdere groepen packages installeren zonder dat dit problemen veroorzaakt. Iedere groepsingang in de *comps* opsomming ziet er ongeveer zo uit:

```
0 Extra Documentation
sag
lpg
howto
faq
man-pages
end
```

Het schijnt dat groepen met een *1* naast de naam (de eerste regel hierboven) standaard voor de installatie worden geselecteerd. Je kunt het Linux installatieproces zelfs verder aanpassen door je eigen groepen aan te maken of bestaande groepen te herdefiniëren door dit bestand te wijzigen.

6.3 Post-installatie shellopdrachten

Dit is waarschijnlijk de beste faciliteit, en iets waarvan geen direct equivalent beschikbaar is in het handmatige installatieproces. Wat we hier kunnen doen is een reeks opdrachten specificeren op niveau van de shell die zouden kunnen worden uitgevoerd nadat de hoofdinstallatie (diskpartitionering, installatie van de packages, enzovoort) compleet is.

Het begin van deze sectie wordt aangeduid door de `%post` directive in het KickStart configbestand. In de opdrachten die daarop volgen kun je voordeel trekken van alle utility's die op je nieuw gebouwde Linux systeem zijn geïnstalleerd, b.v.

```
%post
ln -s /etc/rc.d/init.d /etc/init.d
ln -s /etc/rc.d/rc.local /etc/rc.local
ln -s /usr/bin/md5sum /usr/bin/md5
ln -s /usr/bin/perl /usr/local/bin/perl
chmod ug-s /bin/linuxconf
mkdir /var/tmp/tmp
perl -spi -e 's!image=/boot/vmlinuz-.*!image=/boot/vmlinuz!' /etc/lilo.conf
rm /etc/rc.d/rc*.d/*sendmail
```

Je kunt ook gebruik maken van I/O omleiding en "here"documenten:

```
cat <<EOF >>/etc/passwd
squid:*:102:3500:Squid Proxy:/usr/squid:/bin/bash
EOF

cat <<EOF >>/etc/group
cache:x:3500:
EOF
```

Pas de runtime opstartscripts aan:

```
cat <<EOF >>/etc/rc.local
echo 8192 > /proc/sys/kernel/file-max
echo 32768 > /proc/sys/kernel/inode-max

[ -x /usr/sbin/sshd ] && /usr/sbin/sshd
[ -x /usr/sbin/cfd ] && /usr/sbin/cfd

EOF
```

Stel de *crontab* regels in:

```
cat <<EOF >/tmp/crontab.root
# Keep the time up to date
0,15,30,45 * * * * /usr/sbin/ntpdate -s eggtimer 2>&1 >/dev/null
# Recycle Exim log files
1 0 * * * /usr/exim/bin/exicyclog
# Flush the Exim queue
0,15,30,45 * * * * /usr/exim/bin/exim -q
EOF

crontab /tmp/crontab.root
rm /tmp/crontab.root
```

En installeer zelfs andere RPM's die je zelf hebt samengesteld:

```
rpm -i ftp://chicken.swedish-chef.org/rpms/squid.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/ssh.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/exim.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/cfengine.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/linux.rpm

ssh-keygen -b 1024 -f /etc/ssh_host_key -N ""
depmod -a
```

Je kunt hetzelfde effect bereiken door eigen RPM's aan te maken met de opdrachten die je uit wilt voeren - zie hieronder voor meer informatie. Geef ze een zorgvuldig gekozen naam. Je kunt ze als eerste laten installeren (b.v. naam laten beginnen met 'aaa') of als laatste (b.v. naam laten beginnen met 'zzz').

Een minder moeizame manier om root crontab records aan te maken is ze als bestanden in één of meer van de directory's */etc/cron.hourly*, */etc/cron.daily*, */etc/cron.weekly* en */etc/cron.monthly* aan te maken.

Meer informatie over het samenstellen van je eigen RPM's is beschikbaar in Appendix B.

7 De installatie zelf

Boot zoals gewoonlijk vanaf je RedHat bootdiskette de computer waarop RedHat zal worden geïnstalleerd, maar in plaats van het indrukken van de `RETURN` achter de `SYSLINUX` prompt tik je `linux ks` in.

Als je geluk hebt, is dit alles wat je in hoeft te tikken!

Als je de RedHat bootdiskette zoals hierboven uiteengezet hebt aangepast, hoef je zelfs dit niet te doen :-)

Aangezien we slechs de normale stappen van een RedHat installatie aan het automatiseren zijn, zullen de normale dialogen verschijnen als/wanneer KickStart niet meer weet wat het vervolgens moet doen. In het meest waarschijnlijke geval zal je netwerkinterface niet onmiddellijk automatisch worden gedetecteerd en zal je worden gevraagd naar het IRQ en I/O-adres. KickStart heeft meestal hulp nodig bij ISA-kaarten, en detecteert PCI-kaarten automatisch.

Je kunt zoals gewoonlijk door het wisselen van virtuele consoles in de gaten houden wat KickStart aan het doen is:

- Alt-F1 - installatie dialoog
- Alt-F2 - shellprompt
- Alt-F3 - installatielog (meldingen van het installatieprogramma)
- Alt-F4 - systeemlog (meldingen van de kernel, enz.)
- Alt-F5 - andere meldingen

8 Mounten van de boot/aanvullende disks

De RedHat bootdisk *boot.img* is in het MS-DOS formaat en maakt gebruik van het `SYSLINUX` programma om te booten. De aanvullende disk *supp.img* is een Linux `ext2` bestandssysteem. Je kunt deze bestanden beiden aan je bestandssysteem mounten en ze hacken als je in je Linux-kernel ondersteuning hebt voor het loopback bestandssysteem:

```
# mkdir -p /mnt/boot /mnt/supp
# mount -o loop -t msdos boot.img /mnt/boot
# mount -o loop supp.img /mnt/supp
```

Nu zou je de bestanden op de boot- en aanvullende disk onder respectievelijk de */mnt/boot* en */mnt/supp* moeten kunnen zien. Hè! Mogelijk kunnen oudere versies van `mount` de optie `-o loop` niet hanteren. In deze gevallen moet je expliciet gebruik maken van `losetup` om voor ieder bestand het loopback device te configureren, b.v.

```
# losetup /dev/loop0 boot.img
# mount -t msdos /dev/loop0 /mnt/boot
```

Wellicht dat je expliciet de optie `-t ext2` op moet geven bij het mounten van een `ext2` bestandssysteem zoals hetgeen op de aanvullende disk. Maar het ziet ernaar uit dat mensen met moderne Linux distributies zich hier geen zorgen over hoeven maken.

Als je het niet te moeilijk wilt maken, dan kun je dit altijd omzeilen, en echte diskettes manipuleren in plaats van deze diskette-images.

9 Aanpassen van het RedHat installatieprogramma

Als je wat met de installatieprocedure zelf wilt aanrommelen, de broncode is te vinden op de RedHat CD-ROM of je lokale RedHat mirrorsite. Het bevindt zich in de *misc/src/install* onder de toplevel directory van de *i386* distributie.

Als je de RedHat bootdisk bestudeert, zul je zien dat er naast de Linux kernel *vmlinuz*, een groot bestand *initrd.img* is:

```
- -rwxr-xr-x   1 root     root          559 May 11 15:48 boot.msg
- -rwxr-xr-x   1 root     root          668 May 11 15:48 expert.msg
- -rwxr-xr-x   1 root     root          986 May 11 15:48 general.msg
- -rwxr-xr-x   1 root     root       968842 May 11 15:48 initrd.img
- -rwxr-xr-x   1 root     root         1120 May 11 15:48 kickit.msg
- -r-xr-xr-x   1 root     root         5352 May 11 15:48 ldlinux.sys
- -rwxr-xr-x   1 root     root          875 May 11 15:48 param.msg
- -rwxr-xr-x   1 root     root         1239 May 11 15:48 rescue.msg
- -rwxr-xr-x   1 root     root          402 May 11 15:48 syslinux.cfg
- -rwxr-xr-x   1 root     root       444602 May 11 15:48 vmlinuz
```

Je raadde het al, dit is een ander `ext2` bestandssysteem dat als een bestand is opgeslagen — maar met een handigheidje. Het is in feite ook gecomprimeerd. Je kunt het decomprimeren en het resultaat ervan mounten, b.v.

```
# gzip -dc /mnt/boot/initrd.img >/tmp/initrd.ext2
# mkdir /mnt/initrd
# mount -o loop /tmp/initrd.ext2 /mnt/initrd
```

Waarschijnlijk het belangrijkste onderdeel van dit bestandssysteem is de verzameling laadbare kernelmodules die in de bootdisk is opgenomen. Als je een nieuwe driverversie hierin op wilt nemen, dan moet je of *vmlinuz* door een nieuwe kernel vervangen, welke statisch is gelinkt, of het in de verzameling modules vervangen. Bovendien moet je wellicht andere modules verwijderen om plaats te maken.

Het bestand *modules/modules.cgz* bestaat uit de verzameling modules. Vraag je je af wat dat zou kunnen zijn? In feite is het een gecomprimeerd `cpio` archief, geloof het of niet! En je dacht dat niemand `cpio` nog gebruikte... Zelfs RPM maakt intern nog gebruik van `cpio` Zo kun je er wat mee rommelen:

```
# gzip -dc /mnt/initrd/modules/modules.cgz >/tmp/modules.cpio
# cpio -itv <modules.cpio >modules.listing
# mkdir modules
# cpio -idumv <../modules.cpio
```

Ik denk dat er thans onder Linux geen manier is (tenminste in hoofdstroom distributies) om transparant gecomprimeerde bestandssystemen te benaderen. Laat het me weten als je meer weet!

Denk aan het volgende als je wijzigingen aanbrengt:

1. Maak voor het opnieuw aanmaken van het archief gebruik van `cpio`. Hoe je dit doet wordt als oefening aan de lezer overgelaten...
2. Gebruik `gzip` om het resulterende archief te comprimeren.
3. Kopieer het naar */mnt/initrd*, of waar je je ongecomprimeerde *initrd.img* archief dan ook plaatst.
4. Unmount */mnt/initrd* (of hoe je het ook noemde).
5. Comprimeer het nieuwe *initrd.img* wederom met `gzip`.
6. Kopieer het resulterende archief naar de bootdisk-image - in ons voorbeeld */mnt/boot/initrd.img*.
7. Unmount de bootdisk-image, b.v. */mnt/boot*.

Tenslotte kun je met deze aangepaste bootdisksetup nieuwe bootdiskettes aanmaken, b.v.

```
# cat boot.img >/dev/fd0
```

10 FAQ's/Lijst met wensen

V: Na de KickStart installatie boot mijn computer niet. De BIOS geef een melding als `Missing operating system`.

A: Het lijkt erop dat de partitie met het rootbestandssysteem niet opstartbaar is. Gebruik `fdisk` om naar de opstartbare status over te schakelen.

V: Na de disketteboot, krijg ik de melding: `Error opening files for kickstart copy: File exists`.

A: Gebruik een recentere versie van de *boot.img* en *supp.img* - kijk in de directory *updates* van je lokale RedHat mirror site. Een aantal oudere versies hiervan voor RedHat 5.1 bevatte een fout.

V: Kun je ook alle uitstaande patches (bijgewerkte RPM's) automatisch toepassen? Hoe?

A1: Kopieer de RPM's die je wilt installeren naar de RPMS directory van waaruit de installatie plaats gaat vinden, zie dat je van de oudere RPM's afkomt, en werk het bestand *RedHat/base/hdlist* bij met de nieuwe RPM details. Zie Appendix C voor een script van Eric Doutreleau dat dit voor je doet. Als je het zelf doet, denk er dan aan daarna *genhdlist* uit te voeren!

A2: Probeer dit Perl-script: *patchup* `<http://wwwcache.ja.net/dev/patchup/>`. Het vergelijkt de RPM's die op je systeem zijn geïnstalleerd met die in een voorgestelde directory en rapporteert over degenen waarvan wordt verondersteld dat ze moeten worden bijgewerkt. Ze kunnen zelfs worden geïnstalleerd als je dat het script toevertrouwd.

A3: *rpm2hml* `<http://rufus.w3.org/linux/rpm2html/>` heeft een krachtiger (12MB C code vs. een Perl pagina!) versie van A2.

V: Een enkel config bestand op de installatieserver voor alle clients, misschien als een uitwijkmogelijkheid na het uitproberen van *IPADDR-kickstart* ?

A1: Gebruik de BOOTP/DHCP 'boot file' parameter *bf* om de bestandsnaam in te stellen.

A2: Voeg een record `bf=/kickstart/ks.cfg` toe aan de relevante ingang in */etc/bootptab*.

V: Meer flexibiliteit wanneer er iets mis gaat - b.v. vraag naar de alternatieve lokaties als de distributie niet op de CD-ROM is te vinden.

A: ?

V: Expliet uitsluiten van packages - b.v. alles behalve *sendmail*.

A: Zet het **BASE** package opnieuw op zonder sendmail.

V: Kies welke services automatisch worden gestart bij de systeemstart door de runlevel scripts onder */etc/rc.d/*.

A: Het *chkconfig* utility laat je configureren welke services automatisch tijdens het booten worden uitgevoerd. Je kunt het volgende in je post-installatie scriptsectie uit laten voeren, b.v. voor het uitvoeren van *ypbind* in run levels 3, 4 en 5:

```
chkconfig --level 345 ypbind on
```

en het zal de ypbind level op de 345 level starten.

V:Laat uitvoer op een andere virtuele console verschijnen bij de uitvoering van shellopdrachten in de `%post` sectie, in plaats van het hoofdscherm te overschrijven. *Zou kunnen worden gedaan in de sectie shellopdrachten met `open`?*.

A: Geen probleem - doe bijvoorbeeld dit:

```
exec >/dev/tty5
```

V:Controleert de aanmaakcode voor het bestandssysteem op slechte blokken?

A: Als je naar de virtuele console overschakelt waarop de uitvoer van de aanmaak van het bestandssysteem wordt weergegeven, zie je geen enkele vermelding van de uit te voeren 'read-only' test. Het lijkt erop dat het antwoord 'nee' is.

V:Kan ik het zo regelen dat een aantal machines anders worden geconfigureerd dan anderen?

A: Je zou de host afhankelijke gegevens in de script sectie van de KickStart configuratie kunnen plaatsen - b.v. alleen een gegeven RPM op een gegeven machine installeren. Het zou handig zijn als er een voorwaardelijke installatiefaciliteit in de packages sectie van het configuratiebestand was, b.v. overschakelen bij een andere architectuur, of bij een hostnaam/domein naam/IP-adres.

V:Zijn er wijzigingen tussen RedHat 5.1 en 5.2 ?

A1: Veel wijzigingen in het installatieprogramma, maar voornamelijk correcties op programmeerfouten en cosmetische verbeteringen. Geen impact op KickStart zover als ik je kan zeggen - van een *diff -rcs* van de twee *misc/src/install* directory's.

A2: RH5.2 heeft nu blijkbaar de automatische IP toekenning/DHCP patches voor `bootpd` opgenomen, maar ze hebben de documentatie achterwege gelaten waarin wordt aangegeven hoe je het gebruikt.

V: (Hoe) kun je een specifieke partitie of partities wissen? b.v. */home* behouden maar / legen.

A: Dit kan (nog) niet!

V:Kun je het zo regelen dat je partities op meerdere drives worden aangemaakt? b.v. / op `sda` en */home* op `sdb`.

A: Ik denk van niet - het ziet ernaar uit dat je alleen toegang krijgt tot de eerste drive vanuit het partitionerings hulpmiddel.

V:Is het mogelijk bestaande partities op te laten nemen in de mounttabel, of is het alleen mogelijk de aanmaak van nieuwe partities te specificeren die dan zullen worden opgenomen?

A: ?

V: Waar is het bestand dat wordt aangemaakt bij de uitvoering van `mkkickstart`?

A: Het maakt geen bestand aan - het dumpt de KickStart configuratie naar `stdout`.

V: In de virtuele console 4 (Alt-F4) krijg ik `Unable to load NLS charset cp437(nls_cp437)`. Wat betekent dit? Moet ik me zorgen maken?

A: Het lijkt erop dat je aan het proberen bent een CD-ROM gebrand met het Joliet (Unicode uitbreidingen op ISO 9660) bestandssysteem te mounten. In theorie zijn de bestandsnamen op de CD-ROM wellicht bewerkt en komen ze niet correct over onder Linux. In praktijk schijnt het geen problemen te veroorzaken - zou een onechte afhankelijkheid kunnen zijn?

V:Waarom wordt het X Window Systeem geïnstalleerd? Ik plaatste het niet in mijn lijst met packages.

A: De `XFree86-VGA16` RPM is een 'basiscomponent', en wordt als zodanig altijd geïnstalleerd - tenzij je de definitie van de basisklasse wijzigt.

V: Kan ik in mijn post-installatie script de packages gebruiken die inmiddels zijn geïnstalleerd om geweldige dingen mee uit te halen die niet mogelijk zijn met de beperkte tools op de diskettes?

A: Ja - b.v. als je ervoor kiest Perl te installeren wanneer je de KickStart config samenstelt, is bijna alles mogelijk in ongeveer vijf regels :-)

11 Krediet

Met dank aan Eric Doutreleau voor de info betreft *chkconfig*, de hack voor het configuratiebestand `SYSLINUX`, en het Perlscript voor het bijwerken van de RPM's op de distributieserver. Met dank aan Robert Kaminsky voor het uitgebreide onderzoek. Met dank aan Piete Brooks, Flavia Regina Munhoz, Tom Toffoli, Bob Robbins, Charlie Brady en Ragen Herrington, voor hun opmerkingen en vragen.

12 Appendix A - Configureren van BOOTP/DHCP en NFS

Als je je afvraagt wat in hemelsnaam deze BOOTP en DHCP zooi is, is meer informatie beschikbaar op de *DHCP WWW site* `<http://www.dhcp.org/>`. NFS wordt in de NFS-HOWTO apart gedocumenteerd en er is ook een DHCP mini-HOWTO. Ik heb getracht voldoende details aan te leveren zodat je ermee kunt beginnen, maar heb de onderwerpen niet al te diepgaand behandeld - laat het me weten als je denkt dat dit een beetje te veel van het goede is.

In de BOOTP/DHCP + NFS configuratie die we gaan bespreken, zou het KickStart configuratiebestand via NFS te mounten moeten zijn door de machine van waaraf wordt geïnstalleerd */kickstart/IPADDR-kickstart* op de BOOTP/DHCP server, waar *IPADDR* het IP-adres is van de nieuwe machine, b.v. */kickstart/198.168.254.254-kickstart* voor de machine *198.168.254.254*.

Je zou deze lokatie moeten kunnen overschrijven door de `bf` parameter (boot file) in je BOOTP/DHCP response te retourneren. Wellicht dat het zelfs mogelijk is deze via NFS geheel vanaf een andere machine te mounten.

Maak het bestand */etc/exports* aan met een volgende vergelijkbare inhoud om een aantal directory's voor NFS vanaf een bestaande Linux-box te exporteren:

```
/kickstart *.swedish-chef.org(ro,no_root_squash)
/mnt/cdrom *.swedish-chef.org(ro,no_root_squash)
```

Als je het IP-adres dat je met DNS gaat gebruiken niet hebt geregistreerd dan kan het zijn dat wordt aangegeven dat de NFS-server en/of de RPC portmapper het aflegt. Je kunt dit waarschijnlijk voorkomen door de IP-adres/netmask paren in de configuratiebestanden te plaatsen, b.v.

```
/kickstart 198.168.254.0/255.255.255.0(ro,no_root_squash)
```

en in */etc/hosts.allow*:

```
ALL: 194.82.103.0/255.255.255.0: ALLOW
```

Dit komt doordat de meeste Linux-distributies gebruik maken van TCP-wrappers voor een aantal of alle aan NFS gerelateerde daemons voor het toegangsbeheer. De syntax van */etc/exports* kan op andere Unix-varianten anders zijn - de NFS-servers gebundeld met Linux-distributies bieden gewoonlijke een veel breder bereik aan opties dan die met andere Unix-versies worden geleverd.

Als je in je KickStart configuratiebestand een rootwachtwoord opneemt, of via NFS, directory's met gevoelige informatie exporteert, zal je ervoor moeten zorgen dat je deze informatie aan zo weinig mogelijk mensen blootstelt. Dit kun je bewerkstelligen door de NFS exportpermissies zo verfijnd mogelijk in te stellen, b.v. door een bepaalde host of een bepaald te exporteren subnet te specificeren in plaats van een geheel domein. Het is allemaal mooi en simpel als je een speciaal IP-adres voor KickStart installaties vrijhoudt, behalve als je het later moet wijzigen - of de machine opnieuw moet configureren om het IP-adres via BOOTP/DHCP te krijgen.

De meeste NFS-servers vereisen dat je `mountd` en `nfsd` (onder een aantal Unix-versies worden deze opdrachten voorafgegaan door `rpc.`) te kennen geeft dat het bestand */etc/exports* is gewijzigd - gewoonlijk doe je dit door een `SIGHUP` te verzenden. Er is vaak een programma of script genaamd `exportfs` die dit voor je doet, b.v.

```
# exportfs -a
```

Wellicht dat de directory's niet automatisch worden geëxporteerd wanneer NFS na het booten van de machine niet werkend en draaiend was. Probeer te rebooten, of start als root de volgende programma's op:

```
# portmap
# rpc.nfsd
# rpc.mountd
```

Zoals opgemerkt, wordt op een aantal systemen het voorvoegsel `rpc.` niet gebruikt. Onder de meeste moderne Unix-distributies zijn deze programma's te vinden in de directory's */usr/sbin* of */usr/libexec*. Wellicht dat deze zich nog niet in je pad bevinden, b.v. als je `su` gebruikte om *root* te worden. Het programma `portmap` wordt soms ook `rpcbind` genoemd, b.v. onder Solaris, is voor een aantal versies van `nfsd` vereist dat met een opdrachtregelopdracht het aantal uit te voeren kopieën van de server wordt opgegeven, en het kan zijn dat je bemerkt dat je ook een andere daemon met de naam `biod` moet draaien. Het bovenstaande zou voldoende moeten zijn op (bijna?) alle Linux systemen.

Als je gebruikt maakt van de CMU BOOTP server en dynamische adresseringsextensies waarnaar eerder werd gerefereerd, dan zou een voorbeeld */etc/bootptab* record (*/etc/bootptab* is de normale lokatie van het BOOTP/DHCP configuratiebestand) er ongeveer zo uit zien:

```
.dynamic-1:ip=198.168.254.128:T254=0x30:T250="ds=198.168.254.2:
dn=swedish-chef.org:sm=255.255.255.0:gw=198.168.254.1:
dl=0xFFFFFFFF":
```

Hiermee wordt aangegeven IP-adressen dynamisch toe te kennen bij het aantreffen van nieuwe machines, te beginnen bij *198.168.254.128* en verdergaand voor de volgende 48 (de hexidecimale waarde is *30*) adressen. Iedere client zal de waarde *T250* teruggeven. In dit geval wordt hiermee ingesteld:

- de DNS server `ds` op *198.168.254.2*
- de domeinnaam `dn` op *swedish-chef.org*
- de subnetmask `sm` op *255.255.255.0*
- de standaardgateway `gw` op *198.168.254.1*
- de lease lengte `dl` (hoelang het adres geldig is) op "forever"

Er schijnen een aantal andere verseis van deze server in omloop te zijn die de dynamische adressering niet ondersteunen. Hiervoor zou je een opsomming moeten geven van de hardware (typisch de Ethernet MAC) adressen van iedere te-installeren machine in */etc/bootptab*, en de regels zouden er dan ongeveer zo uit zien:

```
bork.swedish-chef.org:ip=198.168.254.128:ha=0000E8188E56:
  ds=198.168.254.2:dn=swedish-chef.org:sm=255.255.255.0:
  gw=198.168.254.1:dl=0xFFFFFFFF":
```

De parameter `ha` correspondeert met het hardware-adres van de computer waarop wordt geïnstalleerd.

13 Appendix B - Je eigen RPM's aanmaken

Het RPM pakketformaat is reeds zeer goed gedocumenteerd, in het bijzonder in het boek *Maximum RPM* van Ed Bailey, wat je kunt downloaden vanaf de *RPM WWW site* `<http://www.rpm.org/>` - tevens beschikbaar via alle goede boekwinkels! Dit zijn slechts een paar aanwijzingen voor mensen die haast hebben.

RPM packages worden samengesteld vanuit een *spec* bestand. Dit bestaat (op een vergelijkbare wijze als het KickStart configbestand) uit voorgeschreven stappen die moeten worden genomen om het package samen te stellen. Er wordt vanuit gegaan dat je het vanaf de bron op moet bouwen, mogelijk voor meerdere platformen, en waarbij wellicht patches voor het compileren moeten worden toegepast. Eenmaal samengesteld en geïnstalleerd zal een binaire RPM worden aangemaakt uit de bestanden en directory's die je opgeeft als zijnde geassocieerd met het package. Het is belangrijk er notitie van te nemen dat RPM er geen idee van heeft welke bestanden en directory's aan een gegeven package zijn gerelateerd - je moet dit opgeven.

Hier is een voorbeeldspecificatie voor een aangepaste RPM van de *Squid WWW cache server* `<http://squid.nlanr.net/>`:

```
Summary: Squid Web Cache server
Name: squid
Version: 1.NOVM.22
Release: 1
Copyright: GPL/Harvest
Group: Networking/Daemons
Source: squid-1.NOVM.22-src.tar.gz
Patch: retry-1.NOVM.20.patch
%description
Dit is slechts een eerste probeersel om de Squid Web Cache in te pakken
voor een eenvoudige installatie op onze RedHat Linux servers

%prep
%setup
%build
configure --prefix=/usr/squid
perl -spi -e 's!#( -DALLOW_HOSTNAME_UNDERSCORES)!$1!' src/Makefile
make

%install
make install

%files
/usr/squid
```

Zo bouw je deze RPM:

```
% mkdir -p SOURCES BUILD SRPMS RPMS/i386
% cp ~/squid-1.NOVM.22-src.tar.gz SOURCES
% cp ~/retry-1.NOVM.20.patch SOURCES
% rpm -ba squid-1.NOVM.22+retry-1.spec
```

Hiermee zal automatisch een subdirectory onder de *BUILD* directory worden aangemaakt, waarin de broncode zal worden uitgepakt en dan de patch zal worden toegepast (voor het patchen zijn een aantal opties beschikbaar - kijk in het boek voor details). Nu zal RPM automatisch het package bouwen door `configure` en vervolgens `make` uit te voeren, het met `make install` installeren, en een momentopname te nemen van de bestanden onder */usr/squid*. Het is dat laatste wat de binaire RPM van de Squid software zal vormen.

We kunnen willekeurige shellopdrachten bij het uitpak-, bouw- en installatieproces invoegen, zoals b.v. de aanroep naar `perl` waarmee één van Squid's parameters tijdens het compileren wordt verfijnd.

De uiteindelijke binaire RPM zal onder de *RPMS* directory in de voor het platform specifieke subdirectory *i386* worden achtergelaten. In dit geval zal het *squid-1.NOVM.22-1.i386.rpm* worden genoemd. De bestandsnaam wordt aangemaakt door de waarden van de volgende parameters uit het spec bestand aaneen te schakelen: `Name`, `Version` en `Release` - plus het hardwareplatform in kwestie, in dit geval *i386*. Probeer dit in gedachten te houden bij het aanmaken van je eigen RPM's, om te vermijden ze al te lange of moeilijke namen te geven!

Het is het ook waard in gedachten te houden dat je RPM's kunt bouwen zonder dat je het gehele softwarepackage opnieuw moet samenstellen, b.v.

```
Summary: Linux 2.0.36 kernel + filehandle patch + serial console patch
Name: linux
Version: 2.0.36+filehandle+serial_console
Release: 1
Copyright: GPL
Group: Base/Kernel
Source: linux-2.0.36+filehandle+serial_console.tar.gz
%description
Dit is slechts een eerste poging in de Linux kernel met patches samen
te stellen voor installatie op onze RedHat Linux servers.

%prep
echo

%setup
echo

%build
echo

%install
echo

%post
/sbin/lilo

%files
/lib/modules/2.0.36
/boot/vmlinuz
```

In dit geval maken we eenvoudigweg een RPM gebaseerd op het */boot/vmlinuz* bestand en de inhoud van de directory */lib/modules/2.0.36*, en voeren */sbin/lilo* uit nadat het package op een doelcomputer is geïnstalleerd. Laat het me weten als je een veel nettere manier weet om het spec bestand te schrijven.

14 Appendix C - Opnemen van je eigen RPM's in de distributie

Hier is Eric's script voor het opnemen van bijgewerkte RPM's in de RedHat distributie:

```
#!/usr/bin/perl
#
```

```
$redhatdir="/cdrom/i386";
$rpmdir="/cdrom/i386/RedHat/RPMS/";
$updatedir="/cdrom/updates/";
@OTHERDIR=($updatedir);
foreach $dir (@OTHERDIR)
        {
        print "update voor $dir\n";
        system(" find $dir -name \"*.rpm\" -exec cp {} $rpmdir \\; ");
        }
chdir($contribdir) || die "peux pas aller dans $contribdir $!\n";
system("chmod -R 755 $redhatdir");
chdir($rpmdir) || die "problem to go in $rpmdir $!\n";
#
# verwijder het oude bestand
#
opendir(DIR,'.');
@package=grep(/\.rpm$/,readdir(DIR));
foreach $file (@package)
        {
        $file =~ /(.*)\-([\d+|\.]+\w*)\-(\d+)\.[i386|noarch].*/;
        $nom=$1;
        $version=$2;
        $buildvers=$3;
        if ($NOM{$nom})
                {
                $version2=$VERSION{$nom};
                $buildver2=$BUILDVERS{$nom};
                $file2=$FILE{$nom};
                $nom2=$NOM{$nom};
                if ( $version2 gt $version )
                        {
                        print "$file2 is nieuwer dan $file\n";
                        unlink($file);
                        }
                else
                        {
                        if ( $version2 lt $version )
                                {
                                print "$file is nieuwer dan $file2\n";
                                unlink($file2);
                                $VERSION{$nom}=$version;
                                $BUILDVERS{$nom}=$buildvers;
                                $FILE{$nom}=$file;
                                $NOM{$nom}=$nom;
                                }
                        else
                                {
                                if ( $buildver2 > $buildvers )
                                        {
                                   print "$file2 : $buildver2 est mieux que $file : $buildvers\n";
                                   unlink($file);
                                        }
                                else
                                        {
                                print "$file2 : $buildver2 is ouder dan $file : $buildvers\n";
```

```
                                                unlink($file2);
                                                $VERSION{$nom}=$version;
                                                $BUILDVERS{$nom}=$buildvers;
                                                $FILE{$nom}=$file;
                                                $NOM{$nom}=$nom;
                                                    }
                                    }
                            }
                    }
            else
                    {
                    $VERSION{$nom}=$version;
                    $BUILDVERS{$nom}=$buildvers;
                    $FILE{$nom}=$file;
                    $NOM{$nom}=$nom;
                    }
            }

    # hier doen we het zware werk
    #
    system("$redhatdir/misc/src/install/genhdlist $redhatdir");
```