



door Tjabo Kloppenburg
(homepage)

Over de auteur:

Ik raakte geïnfecteerd in 1996. Slackware 3.1 :-). Ik hou van scripten in talen als Python, Perl, GAWK enzovoort. Ook ben ik geïnteresseerd in het booten van apparaten.

Vertaald naar het Nederlands door:
Guus Snijders
<ghs(at)linuxfocus.org>

8cm-Multiboot-CDROM met een aangepaste Knoppix Linux



Kort:

Na het lezen van een paar LinuxFocus artikelen en een paar andere web pagina's, heb ik eindelijk een betrouwbare manier gevonden voor het maken van bootable (Nederlands: opstartbare) cdroms voor het opstarten van schone bootdisks (zonder virussen), single-disk Linux diskettes of zelfs een aangepast Knoppix systeem. Gebrand op een 8cm cdrom met 183mb. Past in iedere zak, en kan behulpzaam zijn in vele situaties.

Benodigheden

Ik ga er van uit dat je over enige basiskennis van Linux beschikt, en dat je weet hoe je CDs moet branden. Het hebben van een brander die ook CD-RW aan kan, kan handig zijn, daar het maken van fouten de manier is waarop we leren... Wat je nodig hebt zijn een CD-RW, een paar 8cm CD-Rs (of CD-RW), en een draaiend Linux systeem met een paar honderd MB vrij. Later, -bij het integreren van Knoppix- zouden we een snelle Internet verbinding moeten hebben, omdat we een paar pakketten van Internet moeten installeren. Als je weet hoe je NFS of SAMBA moet opzetten voor een lokale Debian mirror, heb je de snelle Iternet verbinding niet nodig. Ten slotte zou je over enige kennis van Debian moeten beschikken (installeren en verwijderen van pakketten), of een kennis / IRC kanaal om vragen te stellen. En weten hoe je een loopback device kunt gebruiken.

De Basis

Het opstarten van cdrom heeft veel weg van het opstarten van een floppy of harde schijf. Iets op de schijf wordt door de BIOS in het geheugen gelezen en vervolgens uitgevoerd. In de begin dagen van de cdroms, waren deze niet bedoeld om van op te starten, dus moesten de hardware ontwikkelaars enige magie beoefenen: na het creëren van een paar leveranciers-afhankelijke BIOS uitbreidingen definiëerden ze uiteindelijk de "el torito" standaard.

Deze definiëert een structuur op de cdrom met een bootdisk image, om een floppy disk in het geheugen te emuleren. Na het laden van de disk in het geheugen, boot deze als iedere andere bootdisk.

Met deze floppy emulatie kunnen we een bootable cdrom bouwen met een enkel image van een boot disk, 1.44MB of 2.88MB. Waarschijnlijk heb je nog nooit zo'n floppy gezien, maar de BIOS kan er mee overweg. Tijdens het booten vanaf cdrom hebben geen 2.88MB disk nodig - slechts een disk image gecreëerd met het loop device en een paar tools volstaat.

Een enkele disk is nog geen multi-boot disk. We hebben een tool nodig om andere boot images op het filesysteem te starten. Boot-scriptor is onze vriend. Met boot-scriptor kunnen we vanaf cdrom starten met een loader (zonder disk emulatie, ook dat is mogelijk), en dan kunnen we een disk image in een menu selecteren. Boot-scriptor zorgt vervolgens voor wat magie om dit image in een floppy disk emulatie te laden, vanwaar de image zal starten alsof het een diskette in de floppydrive was.

Deze bootdisks kunnen images zijn van diskettes als windows bootdisks van verschillende versies, een NT wachtwoord veranderaar, mini Linux distributies, of een knoppix bootdisk om een middelgroot Knoppix systeem te starten.

Genoeg gepraat, laten we beginnen

We hebben een directory nodig met genoeg vrije ruimte om de directory structuur van onze cdrom te bouwen. Ik neem aan dat we voldoende vrije ruimte hebben in `"/data"`. De base directory van ons project zal dan bijvoorbeeld `"/data/mboot"` zijn. Hierin creëren we twee directories `"toolcd/"` (inhoud van de cdrom) en `"/archive"` (originele versies van de gebruikte tools, zoals bijvoorbeeld virusscanners).

Boot menu met boot scriptor

Het archief en de INI files zijn te vinden op Boot-Scriptor (bootscriptor.org). Er bevindt zich geen documentatie in het archief, deze is te vinden op de website.

Boot-Scriptor heeft een eigen directory `"bscript/"` nodig op de cdrom, met de loader, een optioneel grafisch bestand (zie de docs), en een bestand genaamd `"bscript.ini"`. Dit bestand definiëert het boot menu in een taal die, net als basic, commando's als `"print"`, `"onkey"` (voor het indrukken van toetsen), en `"memdisk"` (bootdisk image) gebruikt.

Bekijk dit korte voorbeeld eens, en schrijf je eigen ini bestand:

```
print "\ac\c0e--=> my multi boot cdrom with linux <---"  
print "F1 Bootdisk1"  
print " w warm reboot"  
  
MenuLoop:  
onkey f1 goto bootDisk1  
onkey w reboot warm  
goto MenuLoop  
  
bootDisk1:  
memdisk \images\boot1.img
```

Er zijn nog veel meer commando's. Maak een lokale kopie van de boot scriptor documentatie, en plaats deze ergens in de directory structuur van de cdrom. Deze zul je nog nodig hebben bij je eerste stappen in het begrijpen van de basis commando's.

Het maken van de bootdisk images

Voor een eerste test cdrom hebben we slechts een bootdisk nodig, bijvoorbeeld een windows bootdisk. Of een Linux bootdisk, gecreëerd met LILO. Of probeer eens een van die kleine single-disk Linux systemen als HAL of TomsRtBt (2.88MB). Check je favoriete zoekmachine op het web voor meer informatie.

We gebruiken "dd" om een image bestand te maken van de disk:

```
( insert disk )  
# cd /data/mboot/toolcd  
# mkdir images          (directory for the disk images)  
# dd if=/dev/fd0 of=images/boot1.img
```

Het ISO-bestand maken en branden.

Het programma "mkisofs" moet aangeroepen worden met de "-b" optie om er zeker van te zijn dat de bootloader correct wordt geïnstalleerd. Het pad van het argument "-b" is relatief ten opzichte van de basisdirectory van de cdrom ("toolcd"):

```
# cd /data/mboot  
# mkisofs -r -o iso.01 -b bscript/loader.bin \  
          -no-emul-boot -boot-load-size 4 toolcd
```

Nu hoeft er nog maar een ding te gebeuren: het iso bestand op een CD-RW branden. Je kunt hem natuurlijk ook op CD-R branden. Maar slechts een of twee foute karakters in het bestand

"bscript.ini" kunnen je verse cdrom tot de prullenbak verbannen. Dus is het handiger om een CD-RW te gebruiken :-).

Als je een gebruikte CD-RW hebt, moet je deze eerst opschonen. "cdrecord" kent hiervoor twee modi: "blank=fast" en "blank=all". De fast modus is sneller :-).

Alvorens te branden, dien je eerst de dev parameter van je brander te detecteren. Type "cdrecord -scanbus" om alle beschikbare apparaten te zien. Als er geen device is, moet je waarschijnlijk een device driver module als "ide-scsi" laden met een commando als "modprobe ide-scsi".

Probeer het nu nog eens. Hier is een voorbeeld over hoe je het device kunt achterhalen (hier: 0,0,0) en het iso bestand naar een CD-RW branden:

```
# cdrecord -scanbus
Cdrecord 1.10 (i686-pc-linux-gnu) (C) 1995-2001 J. Schilling
Linux sg driver version: 3.1.22
Using libscg version 'schily-0.5'
scsibus0:
  0,0,0  0) 'LG      ' 'CD-RW CED-8083B ' '1.05' Removable CD-ROM
  ...
# cdrecord -v dev=0,0,0 speed=32 iso.01
```

Sommige CD-RWs hebben lage snelheden (bijv.: 4). Maar dat maakt niet uit, "cdrecord" leest de media beschrijvings data van de CD-RW en zal automatisch de hoogst mogelijke snelheid van de media gebruiken of de gegeven waarde (32).

Als het brandproces compleet is, kun je een poging wagen, en booten vanaf de cdrom.

Een Makefile gebruiken

De aanroep van "mkisofs" bestaat uit veel karakters, waardoor het snel kan gebeuren dat er iets wordt vergeten of misgetypt. "cdrecord" vereist ook correcte parameters, het is dus een goed idee om een script of een Makefile te gebruiken, om zeker te zijn van de juiste parameters. Hier zullen we een Makefile gebruiken voor het "make" commando. Installeer deze op je systeem, als deze nog niet aanwezig is. "make" is een van de commando's die een programmeur zou moeten kennen.

We zullen de Makefile in de cdrom directory structuur plaatsen, en deze linken met de "mboot/" directory. Het is een goed idee om alles wat we nodig hebben voor nieuwe versies, op de cdrom te plaatsen. Dit maakt het mogelijk om een nieuwe, betere cdrom te creëren, zelfs als je de bestanden niet meer op de harde schijf hebt, het is dus gelijk een goede backup... :-).

De Makefile kan vrij simpel zijn. Hier is een voorbeeld:

```
BASE = toolcd
DEV = 0,0,0
SPEED = 4
VERSION = 01
ISO = iso.$(BASE).$(VERSION)
```

```

OPTIONS= -b bscript/loader.bin -no-emul-boot -boot-load-size 4

blank:
    cdrecord -v dev=$(DEV) blank=fast

blankall:
    cdrecord -v dev=$(DEV) blank=all

iso:
    echo "deleting ~ files:"
    @find $(BASE) -name "*~" -exec rm {} \;
    echo "creating iso file:"
    @mkisofs -r -o $(ISO) $(OPTIONS) $(BASE)
    @echo
    ls -al $(ISO)

burn:
    cdrecord -v dev=$(DEV) speed=$(SPEED) $(ISO)

```

Het inspringen is gedaan met de **TAB** toets, **Gebruik hier geen spaties voor!**

Zoals al genoemd, plaatsen we de Makefile in een directory in de cdrom structuur. Een plaats als `"/data/mboot/toolcd/scripts"` is goed. We zullen deze symlinken in `"/data/mboot/Makefile"`:

```

# cd /data/mboot
# ln -s toolcd/scripts/Makefile Makefile

```

Om een nieuwe cdrom te maken na het aanpassen van bepaalde dingen (bootdisk images, virus scanners), hoeven we alleen maar drie simpele commando's uit te voeren in de `"mboot/"` directory:

```

# make blank
# make iso
# make burn

```

Dat is beter, nietwaar?

Meer bootdisks

Je zult meer bootdisks vinden in je diskette dozen en op het world wide web. Maar onthoud dat je nooit zeker weet wat zich op een disk image bevindt die je download vanaf het Internet. Ik denk dat het altijd een goed idee is om zoekmachine als Google te gebruiken, om te zien of iemand anders problemen tegenkwam met een bootdisk van een site. En controleer de disks met een virusscanner.

Ik heb een paar bootdisks genomen van www.bootdisk.com. De disks zijn engelse versies, de meeste met cdrom ondersteuning. Je kunt een virusscanner op de cdrom gebruiken, of een bios update uitvoeren vanaf een andere cdrom. Ik vind vooral het "drdf flash" bootdisk image aardig, welke een minimale bootdisk oplevert met voldoende ruimte voor een bios flasher. Je kunt gewoon "rawrite" (dos) of "dd" (Linux) gebruiken om het image op een diskette te plaatsen en je hebt een bootable diskette voor de

flasher in geen tijd. Ik ben ooit bij een LAN party geweest, en zag hoe lang de Linux gurus nodig hadden om een bootable floppy te vinden voor een BIOS update... :-)

Opmerking: "rawrite" is te vinden op de meeste gangbare Linux distributie cdroms.

Meer inhoud

Met je kleine en handige 8cm cdrom heb je een tool om overal met een schone disk te kunnen booten. Waarom geen virus scanner op de cd plaatsen? Ik heb gebruik gemaakt van "F-Prot" van F-Secure. Ze bieden zowel een Linux als een Dos versie aan om vrij te downloaden.

Het enige probleem is hoe je de virus definities op de cdrom kunt updaten. Ik heb gehoord van 8cm CD-RWs, maar heb deze nog nooit gezien in een winkel. Helaas. Een andere oplossing zou zijn om een minimaal Linux systeem van onze 8cm cdrom te booten en vervolgens nieuwe definitie files te downloaden via het Internet.

Twee van de problemen die ik tegenkwam met f-prot waren:

- Het '-' karakter in 'f-prot' veranderde in een '_' tijdens het mounten van de cdrom,
- Het start script verwacht de binary te vinden op een exotische lokatie.

Mijn snelle oplossing was om de binary te kopiëren naar de naam 'fprot', en het script te vergeten :).

Het volgende deel van dit artikel gaat over het integreren van een Linux distributie op onze cdrom, om zo een krachtige tool op te leveren voor bijna iedere paniek situatie.

Knoppix Integreren

Als je nog niet bekend bent met Knoppix, kun je er nu een blik op werpen! Het is een compleet, bootable Linux systeem met 2-3 GB aan software op een enkele 650MB live cdrom. Inclusief hardware detectie en andere nuttige features. Maar het past niet op onze mini cdrom.

Geen nood, het is mogelijk om de knoppix cdrom (of een soortgelijk project) te 'remasteren'. Je hoeft alleen over wat basiskennis van Debian te beschikken, en nog wat meer ruimte op je harde schijf.

Knoppix gebruikt wat speciale magie om 2-3 GB software op een 650MB cdrom te krijgen: het bestandssysteem is gecompriemd in het bestand "/KNOPPIX/KNOPPIX", en wordt gemount door het initiële init proces van een bootende floppy disk (*) met een speciale "cloop" driver module. (* In feite is het de inhoud van een bootdisk image, gebruikt bij het aanroepen van "mkisofs" met de "-b" optie.)

Het bootdisk image dat gebruikt is om de Knoppix cdrom te creëren, kan gevonden worden in de "/KNOPPIX" directory van iedere Knoppix cdrom: "boot.img". Hier zien we het idee weer om alle genen voor de volgende evolutionaire stap "aan boord" te bewaren...

Ik stel voor om de Knoppix bootdisk image te mounten met het loop device om te proberen te begrijpen hoe het boot proces van Knoppix (of Linux) werkt. Zoek op het web naar de "bootdisk howto" voor meer informatie. Een van de bestanden die je zult aantreffen in de disk image, is een ".gz" gecomprimeerd bestand met een bestandssysteem image erin. Dit is de initiële ramdisk met het ramdisk bestandssysteem dat gebruikt wordt op de vroege momenten van het booten.

Tijdens het booten van Knoppix gebeurt dus dit: Het "boot.img" bestand in het iso bestand laadt de Linux kernel van de geëmuleerde floppy disk, en laadt de ramdisk. Het linuxrc proces laadt de "cloop.o" driver module, en zoekt naar een bestand "/KNOPPIX/KNOPPIX" op alle aangesloten apparaten met een bekend bestandssysteem. Inderdaad, het is ook mogelijk om een gecomprimeerde Knoppix vanaf de harde schijf te starten, maar dit heb ik nog niet getest.

Als het apparaat met het gecomprimeerde bestandssysteem is gevonden, wordt deze gemount, en de hardware detectie en alle andere dingen worden gestart.

De floppy disk "boot.img" geeft niks om waar vanaf hij boot (floppy drive, geëmuleerd), dus is het vrij simpel te integreren met onze multi boot cdrom: kopiëer "boot.img" in de "images/" directory, en voeg een hotkey toe aan "bscript.ini".

Knoppix aanpassen

Een originele knoppix cdrom is veel te groot voor onze handige 8cm cdrom. Denk er aan: we hebben 183 MB op een mini cdrom. We zullen vele dingen moeten verwijderen, maar dat is teveel werk. We kunnen ook gebruik maken van het werk van anderen. De website Knoppix Customizations bevat verwijzingen naar een aantal gemodificeerde Knoppix versies, sommige geminimaliseerd tot een gecomprimeerde grootte rond de 50-60 MB. Ik besloot gebruik te maken van "Model_k". Deze is klein, bevat geen GUI, en de ondersteuning voor niet-Engelse toetsenborden ontbreekt. Misschien dat "Damn Small Linux" beter is...

De instructies die je hieronder kunt vinden, komen van een document van Sunil Thomas Thonikuzhiyil, waarin het remastering proces gedetailleerd wordt beschreven. Als je wilt weten hoe je de boot kernel of de initiële ramdisk moet aanpassen, is dit een aanrader. Maar meestal hebben we dat allemaal niet nodig als we "Model_k" of een vergelijkbaar systeem gebruiken voor onze mini cdrom...

Het basis idee achter het remasteren van knoppix ligt in het feit dat knoppix een volledig werkend Debian systeem is - als er een beschrijfbaar bestandssysteem is [een cdrom is read-only :)]. Het belangrijkste deel van het remasteren is de installatie en de-installatie van packages. Niet veel magie hier.

Om een Knoppix systeem te veranderen, moeten we eerst opstarten. Daarna zijn de volgende stappen nodig:

- mount een data ruimte of partitie,
- kopiëer alle ongecomprimeerde bestanden van het met cloop gemounte bestandssysteem,
- chroot naar het ongecomprimeerde systeem,
- mount /proc
- installeer of verwijder software,

- `umount /proc`,
- verlaat de chroot omgeving,
- creëer een nieuw gecomprimeerd bestandssysteem.

Het nieuwe gecomprimeerde bestandssysteem moet gekopieerd worden naar `"/KNOPPIX/KNOPPIX"` in onze cdrom directory structuur, zodat het knoppix boot proces deze zal vinden.

Boot Knoppix en...:

Mount een data ruimte en kopieër bestanden:

```
# mkdir /1
# mount -t ext2 /dev/hda<n> /1
# cp -Rp /KNOPPIX /1
```

Ga naar de ongecomprimeerde, schrijfbaar omgeving:

```
# chroot /1/KNOPPIX
# mount -t proc /proc proc
```

Configureer het netwerk:

```
# (gebruik ifconfig als er geen DHCP server op je netwerk aanwezig is.)
# (verander de nameserver setup als er geen DHCP is. Verwijder de link
  "/etc/resolv.conf" en maak je eigen bestand.)
```

installeer / verwijder packages:

```
# apt-get install joe (Model_k 1.2 komt zonder editor :) )
# ...
```

Wanneer klaar, verlaat het systeem:

```
# (als je /etc/resolv.conf veranderde, verwijder deze en voeg de link
  toe.)
# umount /proc
# exit
```

Nu moet het ge-unmounted bestandssysteem nog gecomprimeerd worden. Het boot proces mount `"/KNOPPIX/KNOPPIX"` met `"cloop"`. We hebben reeds kennis gemaakt met `"loop"` - deze gebruiken we om een bestand met een bestandssysteem te mounten. - `"cloop"` is een `"loop"` met (de)compressie, dus moeten we een bestandssysteem maken met `"mkisofs"`, en deze comprimeren met een speciaal Knoppix programma: `"create_compressed_fs"`:

```
# mkisofs -R /1/KNOPPIX | create_compressed_fs - 65536 > /1/KNOPPIX.2
```


Zoals je kunt zien, wordt de output van "mkisofs" via pipe omgeleid naar de compressie tool. Maak een kopie van je originele "KNOPPIX" bestand, en kopiëer het resulterende bestand "KNOPPIX.2" over "KNOPPIX/KNOPPIX" in de cdrom directory structuur. Maak en brand dan een andere cd en probeer je "nieuwe" knoppix variant uit.

Conclusie

We hebben gezien dat het niet moeilijk is om een handige multi boot cdrom te creëren met handige tools die ons helpen in verschillende situaties. We hebben niet ieder wiel uitgevonden, maar maakten gebruik van wielen die door anderen waren uitgevonden, met wat toegevoegde magie.

We hebben alles wat we nodig hadden op die cdrom, dus hebben we een backup bij de hand - en het is mogelijk om de cdrom te remasteren met alleen de data op de cdrom.

Ik hoop dat plezier hebt beleefd aan mijn korte artikel.

Tot ziens!

Site onderhouden door het LinuxFocus editors team © Tjabo Kloppenburg "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Vertaling info: de --> -- : Tjabo Kloppenburg (homepage) de --> en: Tjabo Kloppenburg (homepage) en --> nl: Guus Snijders < ghs(at)linuxfocus.org >
---	---