De Loopback Root FileSystem HOWTO

door Andrew M. Bishop, *amb@gedanken.demon.co.uk*,
Vertaald door: Ellen Bokhorst, *bokkie@nl.linux.org* v1.1, 24 september 1999

In deze HOWTO wordt uitgelegd hoe het Linux loopback device kan worden gebruikt voor een installatie in het Linux native bestandssysteemformaat dat zonder opnieuw te partitioneren vanaf een DOS-partitie kan worden gedraaid. Tevens worden andere gebruiken met dezelfde techniek besproken.

Inhoudsopgave

1 Introductie

1.1 Copyright

1.2 Revisie Historie

Versie 1.0.0

Initiële Versie (juni 1998)

Versie 1.0.1-1.0.3

Kleine wijzigingen, wijzigingen met betrekking tot het wijzigen van de kernelversie, typfoutjes enz. (1998 - juli 1999)

Versie 1.1

Copyrightinformatie toegevoegd en opnieuw aangeleverd (september 1999)

2 Grondbeginselen van Loopback Devices en Ramdisks

Als eerste zal ik een aantal van de algemene grondbeginselen beschrijven die worden gebruikt bij het instellen van een loopback filesystem als het root-device.

2.1 Loopback Devices

Een **loopback device** onder Linux is een virtueel device dat net als ieder ander media-device kan worden gebruikt.

Voorbeelden van gewone media-devices zijn harddisk-partities zoals `/dev/hda1`, `/dev/hda2`, `/dev/sda1`, of gehele disks zoals het diskettestation `/dev/fd0` enz. Het zijn allen devices die kunnen worden gebruikt om bestanden en directory-structuren te bevatten. Ze kunnen met het benodigde bestandssysteem (ext2fs, msdos, ntfs enz.) worden geformatteerd en vervolgens worden gemount.

Het loopback filesystem associeert een bestand op een ander bestandssysteem als een compleet device. Dit kan dan net als enig ander device worden geformatteerd en gemount. Om dit te kunnen doen wordt het device genaamd `/dev/loop0`, `/dev/loop1`, enz. geassocieerd met het bestand en vervolgens wordt dit nieuwe virtuele device gemount.

2.2 Ramdisk Devices

Onder Linux is ook een ander type virtueel device gemount als bestandssysteem mogelijk, dit is het **ramdisk device**.

In dit geval refereert het device niet naar fysieke hardware, maar naar een deel van het geheugen dat voor dit doel is gereserveerd. Het in beslag genomen geheugen wordt nooit naar disk geswapt, maar blijft in de diskcache.

Een ramdisk kan ten alle tijden worden aangemaakt door naar het ramdisk-device `/dev/ram0` of `/dev/ram1` enz. te schrijven. Dit kan vervolgens op dezelfde manier worden geformatteerd en gemount als het loopback-device.

Wanneer een ramdisk wordt gebruikt om vanaf te booten (zoals vaak op Linux-installatiedisks of rescue-disks wordt gedaan), dan kan het disk-image (de gehele inhoud van de disk als een enkel bestand) op de bootdisk in gecomprimeerde vorm worden opgeslagen. Dit wordt automatisch door de kernel herkend wanneer het boot en het wordt voordat het wordt gemount naar ramdisk gedecomprimeerd.

2.3 Het Initiële Ramdisk Device

Het **initiële ramdisk** device is onder Linux een ander belangrijk mechanisme dat we nodig hebben om een loopback-device als een root-bestandssysteem te kunnen gebruiken.

Wanneer de initiële ramdisk wordt gebruikt, wordt het bestandssysteem image naar het geheugen kopieerd en gemount zodat de bestanden erop kunnen worden benaderd. Een programma op deze ramdisk (genaamd `/linuxrc`) wordt uitgevoerd en wanneer het daarmee klaar is, wordt een ander device als het root-bestandssysteem gemount. De oude ramdisk is echter nog steeds aanwezig en is gemount op de directory `/initrd` als dat er is of beschikbaar via het device `/dev/initrd`.

Dit is ongebruikelijk gedrag aangezien de normale bootreeks vanaf de toegewezen root-partitie boot en het blijft draaien. Met de initiële ramdisk-optie is het mogelijk dat de root-partitie kan wijzigen voordat de hoofd-bootreeks is gestart.

2.4 Het Root Bestandssysteem

Het root-bestandssysteem is het device dat als eerste wordt gemount zodat het na het booten verschijnt als de directory genaamd `/`.

Er zijn een aantal complicaties met het root-bestandssysteem die te wijten zijn aan het feit dat het alle bestanden bevat. Bij het booten worden de `rc` scripts uitgevoerd, dit zijn óf de bestanden in `/etc/rc.d` of `/etc/rc?.d` afhankelijk van de versie van het programma `/etc/init`.

Wanneer het systeem is geboot, is het niet mogelijk de root-partitie te unmounten of wijzigen, aangezien alle programma's het in zekere mate zullen gebruiken. Daarom is de initiële ramdisk zo nuttig, omdat het zo kan worden gebruikt dat de uiteindelijke root-partitie niet dezelfde is als die tijdens het booten wordt geladen.

2.5 De Linux Boot Reeks

Om te laten zien hoe de initiële ramdisk in de bootreeks opereert, is de volgorde van gebeurtenissen hieronder weergegeven.

1. De kernel wordt in het geheugen geladen, dit wordt bewerkstelligd door `LILO` of `LOADLIN`. Je ziet de melding `Loading...` als dit gebeurt.

2. De ramdisk-image wordt in het geheugen geladen, weer wordt dit door `LILO` of `LOADLIN` uitgevoerd. Je ziet de melding `Loading...` als dit gebeurt.

3. De kernel wordt geïnitialiseerd, inclusief de ontleding van de commandoregel-opties en het instellen van de ramdisk als het root-device.

4. Het programma `/linuxrc` wordt op de initiële ramdisk uitgevoerd.

5. Het root-device wordt gewijzigd zoals gespecificeerd in de kernelparameter.

6. Het init-programma `/etc/init` wordt uitgevoerd wat de gebruikersconfigurabele bootreeks uit zal voeren.

Dit is slechts een vereenvoudigde versie van wat er plaatsvindt, maar het is voldoende om uit te leggen hoe de kernel opstart en waar de initiële ramdisk wordt gebruikt.

3 Hoe een Loopback Root Device aan te maken

Nu dat de algemene grondbeginselen zijn uitgelegd, kan de te gebruiken methode voor het aanmaken van het loopback-device worden uiteengezet.

3.1 Benodigdheden

Om het loopback rootdevice aan te maken zijn een aantal dingen benodigd.

- Een werkend Linux-systeem.
- Een manier om grote bestanden naar de DOS-doelpartitie te kopiëren.

Het belangrijkste is de toegang tot een geïnstalleerd Linux-systeem. Dit omdat het loop-device alleen onder Linux kan worden aangemaakt. Dit betekent dat het niet mogelijk is een werkend systeem vanuit het niets te booten. De benodigdheden van het Linux-systeem waar je gebruik van maakt is dat je er een kernel op kunt compileren.

Als het loopback-device éénmaal is aangemaakt, zal het een groot bestand zijn. Ik heb bestanden van 80 MB gebruikt, maar ook al was dit voldoende voor een X-terminal, mogelijk dat het niet genoeg is als je het voor veel meer wilt gebruiken. Dit bestand moet naar de DOS-partitie worden gekopieerd, dus er moet óf van een netwerk, óf van een heleboel diskettes gebruik worden gemaakt.

De benodigde software bestaat uit:

- `LOADLIN` versie 1.6 of hoger
- Een versie van `mount` welke loopback-devices ondersteunt
- Een kernelversie welke de benodigde opties ondersteunt.

Dit zou voor recente Linux-installaties standaard aanwezig moeten zijn.

3.2 Aanmaken van de Linux Kernel

Ik maakte het loopback-device met Linux-kernel versie 2.0.31, ook andere versies zouden moeten werken, maar op z'n minst moeten de volgende opties opgenomen zijn.

De kernel-opties die je zal moeten activeren zijn:

- RAM disk support (`CONFIG_BLK_DEV_RAM`).
- Initial RAM disk (initrd) support (`CONFIG_BLK_DEV_INITRD`).
- Loop device support (`CONFIG_BLK_DEV_LOOP`).
- fat fs support (`CONFIG_FAT_FS`).
- msdos fs support (`CONFIG_MSDOS_FS`).

De eerste twee zijn voor het RAM-diskdevice zelf en voor het initiële ramdisk-device. De volgende is de optie voor het loopback bestandssysteem. De laatste twee bestaan uit de ondersteuning voor het msdos bestandssysteem welke nodig is voor het mounten van de DOS-partitie.

Een kernel compileren zonder modules is de eenvoudigste optie, alhoewel het mogelijk zou moeten zijn als je modules wilt, ik heb het echter niet geprobeerd. Als er modules worden gebruikt, dan zou je ervoor moeten zorgen dat je de bovenstaande opties hebt meegecompileerd en niet als modules.

Mogelijk moet je afhankelijk van de door jouw gebruikte kernel een kernelpatch toepassen. Het is een zeer eenvoudige patch die het je mogelijk maakt het loopback-device als root-bestandssysteem te gebruiken.

- Kernelversies voor 2.0.0; hierover heb ik geen informatie.
- Kernelversie 2.0.0 tot 2.0.34; je moet een kernelpatch toepassen voor 2.0.x kernels zoals hieronder wordt getoond.
- Kernelversie 2.0.35 tot 2.0.x; een kernelpatch is niet nodig.
- Kernelversie 2.1.x; je moet een kernelpatch toepassen voor 2.0.x of 2.2.x kernels zoals hieronder wordt getoond, afhankelijk van de exacte 2.1.x versie.
- Kernelversie 2.2.0 tot 2.2.10; je moet de kernelpatch toepassen voor 2.2.x kernels zoals hieronder getoond.
- Kernelversie 2.3.x; je moet de kernelpatch toepassen voor 2.2.x kernels zoals hieronder getoond.

Bij 2.0.x kernels moet er in het bestand `/init/main.c` een enkele regel worden toegevoegd zoals is weergegeven in de hieronder gewijzigde versie. De regel met `"loop", 0x0700` is de regel die werd toegevoegd.

```
static void parse_root_dev(char * line)
{
        int base = 0;
        static struct dev_name_struct {
                const char *name;
                const int num;
        } devices[] = {
                { "nfs",     0x00ff },
                { "loop",    0x0700 },
                { "hda",     0x0300 },

...
```

```
                { "sonycd",  0x1800 },
                { NULL, 0 }
        };

...

}
```

Bij 2.2.x kernels moeten aan het bestand `/init/main.c` drie regels worden toegevoegd zoals door de gewijzigde versie hieronder wordt getoond. De regel waarin staat `"loop", 0x0700` en de regel ervoor en erna zijn degenen die moeten worden toegevoegd.

```
static struct dev_name_struct {
        const char *name;
        const int num;
} root_dev_names[] __initdata = {
#ifdef CONFIG_ROOT_NFS
        { "nfs",     0x00ff },
#endif
#ifdef CONFIG_BLK_DEV_LOOP
        { "loop",    0x0700 },
#endif
#ifdef CONFIG_BLK_DEV_IDE
        { "hda",     0x0300 },

...

        { "ddv", DDV_MAJOR << 8},
#endif
        { NULL, 0 }
};
```

Zodra de kernel is geconfigureerd, kan het worden gecompileerd waarmee een bestand `zImage` wordt geproduceerd (`make zImage`). Dit bestand zal na de compilatie te vinden zijn in `arch/i386/boot/zImage`.

3.3 Aanmaken van de Initiële Ramdisk Device

De initiële ramdisk wordt het eenvoudigst van het begin aangemaakt als een loopback-device. Je zal dit als root moeten doen. De commando's die je hiervoor uit zal moeten voeren, staan hieronder. Er wordt vanuit gegaan dat ze vanuit de home-directory van root (`/root`) worden opgestart.

```
mkdir /root/initrd
dd if=/dev/zero of=initrd.img bs=1k count=1024
mke2fs -i 1024 -b 1024 -m 5 -F -v initrd.img
mount initrd.img /root/initrd -t ext2 -o loop
cd initrd
[maak de bestanden aan]
cd ..
umount /root/initrd
gzip -c -9 initrd.img > initrdgz.img
```

Hier zijn een aantal stappen voor, en die kunnen als volgt worden beschreven.

1. Maak een mountpoint voor de initiële ramdisk (een lege directory).
2. Maak een leeg bestand van de benodigde grootte aan. Hier heb ik 1024kB gebruikt, mogelijk heb je afhankelijk van de inhoud meer of minder nodig (de grootte is de laatste parameter).
3. Maak in het lege bestand een ext2 bestandssysteem aan.
4. Mount het bestand op het mountpoint, dit gebruikt het loopback-device.
5. Ga naar het gemounte loopback-device.
6. Maak de benodigde bestanden aan (zie hieronder voor details).
7. Ga uit het gemounte loopback-device.
8. Unmount het device.
9. Maak een comprimeerde versie voor later gebruik.

Inhoud Van De Initiële Ramdisk

De bestanden die je op de ramdisk nodig zult hebben, zijn de minimum vereisten om commando's uit te kunnen voeren.

- `/linuxrc` Het script dat wordt uitgevoerd om het msdos bestandssysteem uit te voeren (zie hieronder).
- `/lib/*` De dynamische linker en library's voor de programma's.
- `/etc/*` De cache die door de dynamische linker wordt gebruikt (niet echt nodig, maar maakt dat het stopt foutmeldingen te produceren).
- `/bin/*` Een shell-interpreter (`ash` omdat het kleiner is dan `bash`. De `mount` en `losetup` programma's voor het afhandelen van de DOS-disk en het instellen van de loopback-devices.
- `/dev/*` De devices die zullen worden gebruikt. Je hebt `/dev/zero` voor `ld-linux.so` nodig, `/dev/hda*` om de msdos-disk te mounten en `/dev/loop*` voor het loopback device.
- `/mnt`Een lege directory om de msdos-disk op te mounten.

De initiële ramdisk die ik gebruikte, vind je hieronder. De inhoud kwam uit op 800kB waarbij de overhead van het bestandssysteem is inbegrepen.

```
total 18
drwxr-xr-x   2 root     root         1024 Jun  2 13:57 bin
drwxr-xr-x   2 root     root         1024 Jun  2 13:47 dev
drwxr-xr-x   2 root     root         1024 May 20 07:43 etc
drwxr-xr-x   2 root     root         1024 May 27 07:57 lib
-rwxr-xr-x   1 root     root          964 Jun  3 08:47 linuxrc
drwxr-xr-x   2 root     root        12288 May 27 08:08 lost+found
drwxr-xr-x   2 root     root         1024 Jun  2 14:16 mnt

./bin:
total 168
-rwxr-xr-x   1 root     root        60880 May 27 07:56 ash
-rwxr-xr-x   1 root     root         5484 May 27 07:56 losetup
-rwsr-xr-x   1 root     root        28216 May 27 07:56 mount
lrwxrwxrwx   1 root     root            3 May 27 08:08 sh -> ash
```

```
./dev:
total 0
brw-r--r--   1 root     root       3,   0 May 20 07:43 hda
brw-r--r--   1 root     root       3,   1 May 20 07:43 hda1
brw-r--r--   1 root     root       3,   2 Jun  2 13:46 hda2
brw-r--r--   1 root     root       3,   3 Jun  2 13:46 hda3
brw-r--r--   1 root     root       7,   0 May 20 07:43 loop0
brw-r--r--   1 root     root       7,   1 Jun  2 13:47 loop1
crw-r--r--   1 root     root       1,   3 May 20 07:42 null
crw-r--r--   1 root     root       5,   0 May 20 07:43 tty
crw-r--r--   1 root     root       4,   1 May 20 07:43 tty1
crw-r--r--   1 root     root       1,   5 May 20 07:42 zero

./etc:
total 3
-rw-r--r--   1 root     root         2539 May 20 07:43 ld.so.cache

./lib:
total 649
lrwxrwxrwx   1 root     root           18 May 27 08:08 ld-linux.so.1 -> ld-linux.so.1.7.14
-rwxr-xr-x   1 root     root        21367 May 20 07:44 ld-linux.so.1.7.14
lrwxrwxrwx   1 root     root           14 May 27 08:08 libc.so.5 -> libc.so.5.3.12
-rwxr-xr-x   1 root     root       583795 May 20 07:44 libc.so.5.3.12

./lost+found:
total 0

./mnt:
total 0
```

De enige complexe stappen hierbij zijn de devices in `dev`. Gebruik het programma `mknod` om ze aan te maken, gebruik de bestaande devices in `/dev` als een template voor het verkrijgen van de benodigde parameters.

Het bestand /linuxrc

Het bestand `/linuxrc` op de initiële ramdisk is nodig voor alle voorbereidingen zodat het loopback-device voor de root-partitie kan worden gebruikt als 't beëindigt.

Het voorbeeld hieronder probeert `/dev/hda1` als een msdos-partitie te mounten en als het daarin slaagt stelt het de bestanden `/linux/linuxdsk.img` als `/dev/loop0` en `/linux/linuxswp.img` als `/dev/loop1` in.

```
#!/bin/sh

echo INITRD: Probeer /dev/hda1 als msdos te mounten

if /bin/mount -n -t msdos /dev/hda1 /mnt; then

   echo INITRD: Mount OK
   /bin/losetup /dev/loop0 /mnt/linux/linuxdsk.img
   /bin/losetup /dev/loop1 /mnt/linux/linuxswp.img
   exit 0

else

   echo INITRD: Mount mislukt
   exit 1
```

```
fi
```

Het eerste device /dev/loop0 zal het root-device worden en het tweede /dev/loop1 device de swap-space.

Als je waneer je klaar bent als niet-root gebruiker naar de DOS-partitie wilt kunnen schrijven, dan kun je in plaats daarvan gebruik maken van `mount -n -t msdos /dev/hda1 /mnt -o uid=0,gid=0,umask=000,quiet`. Hiermee zal alle toegang tot de DOS-partitie naar root worden ingedeeld en zullen de permissies dienovereenkomstig worden ingesteld.

3.4 Aanmaken van het Root Device

Het te gebruiken root-device is het bestand linuxdsk.img. Je zal dit op dezelfde wijze, maar dan groter, aan moeten maken zoals de initiële ramdisk werd aangemaakt. Je kunt iedere gewenste Linux-installatie op deze disk installeren.

De eenvoudigste manier zou kunnen zijn om er een bestaande Linux-installatie naar te kopiëren. Een alternatief is er een nieuwe Linux-installatie op te kopiëren.

Ervan uitgaande dat je dit hebt gedaan, zijn er nog een paar kleine wijzigingen nodig.

Het bestand /etc/fstab moet naar de root-partitie en de swap verwijzen door gebruik te maken van de twee loopback-devices die op de initiële ramdisk zijn ingesteld.

```
/dev/loop0     /      ext2   defaults 1 1
/dev/loop1     swap   swap   defaults 1 1
```

Dit zal ervoor zorgen dat de kernel niet in de war zal raken waar het root-device is als het echte root-device zal worden gebruikt. Bovendien maakt het 't mogelijk de swap-space op dezelfde manier toe te voegen zoals het gewoonlijk wordt gedaan. Je zou iedere andere verwijzing naar een rootdisk-device of swap-partitie moeten verwijderen.

Als je nadat Linux is opgestart de DOS-partitie wilt kunnen lezen, dan zal je nog een aantal extra wijzigingen moeten maken.

Maak een directory genaamd /initrd aan, hier zal de initiële ramdisk worden gemount zodra het loopback root-bestandssysteem is gemount.

Maak een symbolische link genaamd /DOS aan dat verwijst naar /initrd/mnt waar de echte DOS-partitie zal worden gemount.

Voeg een regel toe aan het rc bestand waarmee de disks worden gemount. Hier zal het commando `mount -f -t msdos /dev/hda1 /initrd/mnt` worden uitgevoerd, waarmee een 'nep' mount van de DOS-partitie zal worden aangemaakt, waardoor alle programma's (zoals df) zal weten dat de DOS-partitie is gemount en waar het te vinden is. Als je andere opties gebruikte in het bestand /linuxrc dan zou je die ook hier moeten gebruiken.

Een Linux-kernel op dit root-device is niet nodig, aangezien dat reeds eerder is geladen. Als je echter gebruik maakt van modules, dan zou je ze net als anders op dit device in moeten voegen.

3.5 Aanmaken van het Swap Device

Het root-device dat je zal gaan gebruiken, is het bestand linuxswap.img. Het swap-device is zeer eenvoudig aan te maken. Maak een leeg bestand aan zoals dit werd gedaan voor de initiële ramdisk en start dan `mkswap linuxswap.img` op om het te initialiseren.

De grootte van de swap-space is afhankelijk van wat je van plan bent met het geïnstalleerde systeem, maar ik raad je een hoeveelheid RAM aan tussen 8 MB en de hoeveelheid RAM dat je hebt.

3.6 Aanmaken van de MSDOS-Directory

De bestanden die gebruikt gaan worden, moeten naar de DOS-partitie worden verplaatst.

Dit zijn de bestanden die nodig zijn in de DOS-directory genaamd `C:\LINUX`:

- `LINUXDSK.IMG`De disk-image dat het root-device wordt.
- `LINUXSWP.IMG` De swap space.

3.7 Aanmaken van de opstartdiskette

De opstartdiskette die wordt gebruikt is slechts een gewone DOS geformatteerde opstartdiskette.

Deze wordt vanuit DOS met behulp van `format a: /s` aangemaakt.

Op deze disk zal je een `AUTOEXEC.BAT` aan moeten maken (als hieronder) en de kernel, gecomprimeerde initiële ramdisk en het uitvoerbare bestand `LOADLIN` naar moeten kopiëren.

- `AUTOEXEC.BAT` Het door DOS automatisch uitgevoerde batchbestand.
- `LOADLIN.EXE` Het uitvoerbare programma `LOADLIN`.
- `ZIMAGE` De Linux-kernel.
- `INITRDGZ.IMG` De gecomprimeerde initiële ramdisk-image.

In het bestand `AUTOEXEC.BAT` hoort slechts de volgende regel voor te komen.

```
\loadlin \zImage initrd=\initrdgz.img root=/dev/loop0 ro
```

Hiermee wordt het te gebruiken kernel-image gespecificeerd, de initiële ramdisk-image, het root-device nadat de initiële ramdisk klaar is en dat de root-partitie read-only zal worden gemount.

4 Het booten van het Systeem

Alles wat je nodig hebt om vanaf dit nieuwe root-device te booten, is dat de diskette geprepareerd zoals hiervoor is beschreven in het diskettestation is gedaan om vanaf te booten.

Je zal de volgende reeks gebeurtenisssen te zien krijgen.

1. DOS boot
2. AUTOEXEC.BAT start
3. LOADLIN wordt uitgevoerd
4. De Linux-kernel wordt naar het geheugen gekopieerd
5. De initiële ramdisk wordt naar het geheugen gekopieerd
6. De Linux-kernel is begonnen aan de uitvoering

7. Het bestand `/linuxrc` op de initiële ramdisk wordt uitgevoerd
8. De DOS-partitie is gemount en de root en swap-devices ingesteld
9. De bootreeks continueert vanaf het loopback-device

Wanneer het systeem hiermee klaar is, kun je de opstartdiskette verwijderen en het Linux-systeem gebruiken.

4.1 Mogelijke Problemen Met Oplossingen

Er zijn een aantal fasen waarin dit proces zou kunnen mislukken, ik zal proberen uit te leggen wat dat zijn en wat te controleren.

Wanneer DOS boot is eenvoudig te herkennen door de melding `MS-DOS Starting ...` op het scherm. Als deze melding niet verschijnt, dan is de diskette óf niet opstartbaar of het systeem kan niet worden opgestart vanaf het diskettestation.

Wanneer de commando's in het `AUTOEXEC.BAT` bestand worden uitgevoerd, zouden deze standaard naar het scherm moeten worden geëchoot. In dit geval gaat het slechts om een enkele regel waarmee `LOADLIN` wordt opgestart.

Wanneer `LOADLIN` wordt uitgevoerd, zal het twee zeer zichtbare dingen doen, als eerste zal het de kernel in het geheugen laden, ten tweede zal het de ramdisk naar het geheugen kopiëren. Beiden zijn te herkennen aan een `Loading...` melding.

De kernel begint zichzelf te comprimeren, hierdoor kunnen **crs** fouten ontstaan als de kernel-image beschadigd is. Vervolgens zal het de initilisatiereeks starten, welke zeer woordenrijk met diagnostische meldingen is. Tijdens deze fase is het laden van het initiële ramdisk-device ook zichtbaar.

Tijdens de uitvoering van het bestand `/linuxrc` zijn er geen diagnostische meldingen, maar die kun je zelf als hulp bij het debuggen toevoegen. Als het in dit stadium niet lukt het loopback-device als het root-device in te stellen, dan zie je wellicht een melding dat er geen root-device is en breekt de kernel af.

De normale bootreeks van het nieuwe root-device zal nu verdergaan en hierbij worden heel wat meldingen weergegeven. Er kunnen problemen optreden met het read-write mounten van het root-device, maar de commandoregel-optie '`ro`' van LOADLIN kan dit verhelpen. Andere problemen die op kunnen treden bestaan daaruit dat de bootreeks van slag is over waar het root-device is, dit is waarschijnlijk te wijten aan een probleem met `/etc/fstab`.

Wanneer de bootreeks is voltooid, blijft er nog een probleem over dat programma's van slag zijn of de DOS-partitie wel of niet is gemount. Daarom is het een goed idee het eerder beschreven nep mount commando te gebruiken. Dit maakt 't leven er een stuk eenvoudiger op als je de bestanden op het DOS-device wilt benaderen.

4.2 Refererende Documenten

De documenten die ik gebruikte om mijn eerste loopback root-bestandssysteem aan te maken, waren:

- De Linux kernelsource, in het bijzonder `init/main.c`
- De Linux kerneldocumentatie, in het bijzonder `Documentatie/initrd.txt` en `Documentation/ramdisk.txt`.
- De `LILO` documentatie.
- De `LOADLIN` documentatie

5 Andere Mogelijkheden met het Loopback Root Device

Nu het principe van het booten van een bestandssysteem in een bestand op een DOS-partitie is bewezen, zijn er veel andere dingen die je nu kunt doen.

5.1 DOS Harddisk Installatie

Als het mogelijk is Linux vanaf een bestand op een DOS-harddisk te booten door gebruik te maken van een opstartdiskette, dan is het uiteraard ook mogelijk dit te doen door de harddisk zelf te gebruiken.

Een configuratie bootmenu kan worden gebruikt met de optie om `LOADLIN` vanuit `AUTOEXEC.BAT` uit te voeren. Hierdoor zal de bootreeks sneller zijn, maar verder is het identiek.

5.2 LILO Boot Installatie

Het gebruik van `LOADLIN` is slechts één optie voor het booten van een Linux-kernel. Er is ook `LILO` dat vrijwel hetzelfde doet maar zonder dat het DOS nodig heeft.

In dit geval kan de voor DOS geformatteerde diskette worden vervangen door één die met ext2fs is geformatteerd. De details zijn anders zeer vergelijkbaar, met de kernel en de initiële ramdisk bestanden op die disk.

De reden dat ik koos voor de `LOADLIN` methode is dat de argumenten die aan `LILO` moeten worden meegegeven wat complex zijn. Ook is het vanzelfsprekender voor een terloopse waarnemer waar de diskette voor is, aangezien het onder DOS kan worden ingelezen.

5.3 VFAT / NTFS Installatie

Ik heb de NTFS-methode geprobeerd en had er geen problemen mee. De driver voor het NTFS bestandssysteem is in versie 2.0.x geen standaard kerneloptie, maar het is als een patch beschikbaar vanaf *http://www.informatik.hu-berlin.de/˜loewis/ntfs/*. Onder versie 2.2.x is de NTFS driver standaard in de kernel opgenomen.

De enige wijzigingen voor het VFAT- of NTFS-opties bestaan uit de initiële ramdisk, het bestand `/linuxrc` moet in plaats van msdos een bestandssysteem van het type vfat of ntfs mounten.

Ik zou niet weten waarom dit niet tevens op een VFAT-partitie zou werken.

5.4 Linux zonder herpartitioneren installeren

Voor het proces waarbij Linux vanaf een standaarddistributie op een PC wordt geïnstalleerd, is een diskette nodig en het herpartitioneren van de disk. Deze fase zou in plaats daarvan kunnen worden bewerkstelligd door een bootdiskette, waarmee een leeg loopback-device en een swapbestand worden aangemaakt. Hiermee zou het mogelijk worden de installatie als normaal voort te zetten, maar zou in het loopbackdevice installeren in plaats van op een partitie.

Dit zou als alternatief voor een `UMSDOS` installatie kunnen worden gebruikt, het zou efficiënter in diskgebruik zijn, aangezien de minimum inbeslaggenomen eenheid in het ext2 bestandssysteem 1kB is in plaats van de tot aan 32kB op DOS-partities. Het kan ook op VFAT en als NTFS geformatteerde disks worden gebruikt welke anders een probleem zou zijn.

5.5 Booten van een Niet-opstartbaar device

Deze methode kan ook worden gebruikt om een Linux systeem vanaf een device te booten dat normaal gesproken niet opstartbaar is.

- CD-Rom
- Zip Disks
- Parallelle poort diskdrives

Uiteraard zijn er nog vele andere devices die zouden kunnen worden gebruikt, NFS root filesystems zijn reeds in de kernel als een optie opgenomen, maar in plaats daarvan zou ook de hier beschreven methode kunnen worden gebruikt.