Loop Root Device mini-HOWTO

Andrew M. Bishop (amb@gedanken.demon.co.uk) und Hans-Jürgen Geiß (hans-juergen.geiss@kl.maus.de)

v1.1-2, 25. Oktober 1999

Dieses HOWTO erklärt, wie man unter Linux ein Loop-Device verwendet, um ein Linux-System zu installieren, das auf einer DOS-Partition betrieben werden kann, ohne diese neu zu partitionieren. Andere Anwendungen die auf dieser Technik basieren werden ebenfalls beschrieben.

Inhaltsverzeichnis

1 Copyright

2 Grundsätzliches über Loop-Devices und RAM-Disks

Zuerst will ich einige grundlegende Prinzipien erläutern, die beim Erstellen eines Dateisystems auf einem Loop-Device verwendet werden.

2.1 Loop-Devices

Ein *Loop-Device* ist ein virtuelles Laufwerk, das wie alle anderen Laufwerke unter Linux verwendet werden kann.

Beispiele für gewöhnliche Laufwerke sind: einzelne Festplattenpartitionen wie `/dev/hda1`, `/dev/hda2`, `/dev/sda1` oder komplette Laufwerke wie die Diskettenlaufwerke `/dev/fd0` usw.. Sie alle können verwendet werden, um Dateien oder Verzeichnisse zu speichern. Sie können mit dem benötigten Dateisystem formatiert (Ext2, MSDOS, NTFS usw.) und dann in die Verzeichnisstruktur von Linux eingefügt werden (`mount`).

Das Loop-Device verwendet eine Datei in einem anderen Dateisystem als ein virtuelles Laufwerk, welches dann genauso angesprochen werden kann wie die bereits oben aufgelisteten Medien. Hierfür wird die Gerätedatei `/dev/loop0` oder `/dev/loop1` usw. mit der Datei verbunden und dann dieses neue virtuelle Laufwerk gemountet.

2.2 RAM-Disks

Unter Linux ist es auch möglich, ein anderes virtuelles Laufwerk als Dateisystem zu mounten, die sog. RAM-Disk.

In diesem Fall verweist das Gerät nur auf einen Teil des RAM-Speichers, der hierfür zur Verfügung gestellt wurde. Der zugewiesene Speicher wird nicht auf die Festplatte (Swap Device) ausgelagert. Das Laufwerk wird aber im Cache-Speicher gepuffert.

Eine RAM-Disk kann jederzeit erstellt werden, indem auf die Gerätedatei `/dev/ram0` oder `/dev/ram1` geschrieben wird. Sie kann genau wie ein Loop-Devivce formatiert und gemountet werden.

Wenn eine RAM-Disk verwendet wird, um davon zu booten, wie es oft bei Installations- oder Rescue-Disketten von Linux getan wird, kann das Disk-Image (der komplette Inhalt der Diskette in einer einzelnen Datei) komprimiert auf der Boot-Diskette gespeichert werden. Beim Booten wird dies automatisch vom Kernel erkannt und die Datei auf eine RAM-Disk entpackt, bevor diese dann gemountet wird.

2.3 Die Initial-RAM-Disk

Die *Initial-RAM-Disk* unter Linux ist ein weiterer wichtiger Mechanismus, den wir benötigen, um ein Loop-Device als Root-Dateisystem zu verwenden.

Bei Verwendung einer Initial-RAM-Disk wird zunächst das Disk-Image in den Speicher geladen und dann gemountet, damit die Dateien darauf ausgeführt werden können. Ein Programm, genauer gesagt die Skript-Datei `/linuxrc`, auf der RAM-Disk wird ausgeführt und wenn es fertig ist, wird ein anderes Speichermedium in das Root-Dateisystem gemountet. Die alte RAM-Disk ist aber noch verfügbar und im Verzeichnis `/initrd` gemountet, sofern es vorhanden ist, oder sie ist über die Geräte-Datei `/dev/initrd` erreichbar.

Das ist ein ungewöhnliches Verhalten, weil normalerweise der Boot-Vorgang nur von der vorbestimmten Root-Partition gestartet und weiter ausgeführt wird. Mit der Initial-RAM-Disk kann man die Root-Partition aber ändern, bevor die eigentliche Boot-Sequenz startet.

2.4 Das Root-Dateisystem

Das Root-Dateisystem wird als erstes gemountet, damit es als das Verzeichnis / nach dem Booten erscheint.

Es gibt eine Reihe von Komplikationen beim Root-Dateisystem, die auf die Tatsache zurück zu führen sind, daß es alle Dateien beinhaltet. Beim Booten werden die `rc`–Skripte gestartet. Das sind die Dateien `/etc/rc.d` oder `/etc/rc?.d`, je nach Version des Init-Programmes `/etc/init`.

Wenn das System gebootet hat, ist es nicht mehr möglich, die Root-Partition zu entfernen oder zu wechseln, da alle Programme sie zu einem gewissem Grad verwenden. Darum ist die Initial-RAM-Disk so nützlich, denn sie kann benützt werden, um am Ende des Bootens eine andere Root-Partition zu haben als am Anfang.

2.5 Der Boot-Vorgang bei Linux

Um zu zeigen, wie die Initial-RAM-Disk arbeitet, sind die einzelnen Vorgänge beim Booten unten aufgelistet.

1. Der Kernel wird in den Speicher geladen, was von `LILO` oder `LOADLIN` bewerkstelligt wird. Wenn das geschieht, können Sie die Meldung `Loading...` sehen.
2. Das RAM-Disk-Image wird in den Speicher geladen, wiederum durch `LILO` oder `LOADLIN` und Sie können erneut die Meldung `Loading...` sehen.
3. Der Kernel wird intitialisiert, die Optionen der Kommandozeile werden ausgewertet und die RAM-Disk als Root-Partition aktiviert.
4. Das Programm `/linuxrc` auf der Initial-RAM-Disk wird ausgeführt.
5. Die Root-Partition wird gewechselt auf die mit den Kernel-Parametern angegebene Partition.
6. Das Init-Programm `/etc/init` wird gestartet, welches die durch den Benutzer konfigurierbare Boot-Sequenz durchführt.

Das ist nur eine vereinfachte Darstellung, aber sie reicht aus, um zu erläutern, wie der Kernel gestartet und wo die Initial-RAM-Disk verwendet wird.

3 Wie ein Loop-Root-Device erzeugt wird

Nun, wo die grundlegenden Prinzipien erläutert sind, kann die Erstellung eines Loop-Devices erklärt werden.

3.1 Voraussetzungen

Um ein Loop-Device zu erstellen, werden mehrere Dinge benötigt.

- Ein funktionierendes Linux-System
- Eine Möglichkeit große Dateien auf die Ziel-Partition (DOS-Festplatte) zu schreiben.

Am wichtigsten ist der Zugriff auf ein installiertes Linux-System, denn das Loop-Device kann nur unter Linux erstellt werden. Das heißt, daß es nicht möglich ist, ein funktionierendes System aus dem Nichts zu erstellen (Boot Strapping). Mit dem Linux-System, das Sie verwenden, müssen Sie einen neuen Kernel kompilieren können.

Wenn das Loop-Device erst einmal erzeugt ist, wird es eine sehr große Datei sein. Ich habe Dateien mit einer Größe von 80 MB verwendet. Aber obwohl das ausreichte, um ein X-Terminal zu betreiben, könnte es zu klein sein, wenn Sie es für etwas anderes verwenden möchten. Diese Datei muß auf die DOS-Partition kopiert werden; also müssen entweder ein Netzwerk oder eine große Anzahl von Disketten verwendet werden.

Die benötigten Programme sind:

- `LOADLIN` Version 1.6 oder höher
- Eine Version von `mount` die Loop-Devices unterstützt
- Eine Version des Kernels die die erforderlichen Optionen unterstützt

Das sollte aber bei den jüngeren Linux-Distributionen Standard sein.

3.2 Den Linux-Kernel erzeugen

Ich erzeugte das Loop-Device unter Verwendung des Linux-Kernels 2.0.31. Andere Versionen sollten ebenfalls funktionieren. Sie müssen aber zumindest die unten aufgelisteten Optionen haben.

Folgende Kernel-Optionen werden verwendet:

- 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`).

Die ersten beiden sind für die RAM-Disk und die Intitial-RAM-Disk. Die nächste ist für das Loop-Device. Die letzten beiden sind für Unterstützung des DOS-Dateisystems, das gebraucht wird, um die DOS-Partition zu mounten.

Die einfachste Möglichkeit ist es, einen Kernel ohne nachladbare Module zu erzeugen. Mit Modulen sollte es auch funktionieren, obgleich ich es nicht ausprobiert habe. Wenn Module verwendet werden, sollten Sie sich vergewissern, daß zumindest die o.g. Optionen fest in den Kernel eingebunden und nicht als Module erzeugt werden, damit sie gleich zu Beginn zur Verfügung stehen.

Je nachdem welche Kernel-Version verwendet wird, könnte es sein, daß Sie einen Kernel-Patch anwenden müssen. Es ist eine sehr einfache Anpassung, die es erlaubt, ein Loop-Device als Root-Partition zu verwenden.

Kernel-Versionen älter als 2.0.0

Es liegen mir keine weiteren Informationen darüber vor.

Kernel-Versionen 2.0.0 bis 2.0.34

Sie müssen den Kernel-Patch für die 2.0.x Kernels anwenden, wie unten beschrieben.

Kernel-Versionen 2.0.35 bis 2.0.x

Ein Kernel-Patch ist nicht erforderlich.

Kernel version 2.1.x

Sie müssen den Kernel-Patch für die 2.0.x-Kernels oder 2.2.x-Kernels anwenden, je nach genauer Version des 2.1.x-Kernels, wie unten beschrieben.

Kernel-Versionen 2.2.0 bis 2.2.10

Sie müssen den Kernel-Patch für die 2.2.x-Kernels anwenden, wie unten beschrieben.

Kernel-Version 2.3.x

Sie müssen den Kernel-Patch für die 2.2.x-Kernels anwenden, wie unten beschrieben.

Für die 2.0.x-Kernels muß eine einzelne Zeile in die Datei `init/main.c` eingefügt werden. Die einzufügende Zeile lautet:

```
{ "loop", 0x0700 }
```

Das sollte dann so aussehen:

```
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 }
        };

...

}
```

Für die 2.2.x-Kernels müssen drei Zeilen in die Datei `init/main.c` eingefügt werden. Die eingefügten Zeilen sind

```
{ "loop", 0x0700 }
```

und die Zeilen die davor und dahinter stehen:

```
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 }
};
```

Wenn der Kernel konfiguriert ist, sollte er übersetzt werden, um eine `zImage`-Datei zu erhalten (`make zImage`). Nach dem Übersetzen wird die entstandene Datei `arch/i386/boot/zImage` heißen.

3.3 Die Initial-RAM-Disk erstellen

Die Initial-RAM-Disk wird für den Startvorgang meistens als ein Loop-Device angelegt. Sie müssen das als Systemverwalterin (Super-User) tun, weil es Root-Rechte erfordert. Die Programme, die dazu ausgeführt werden müssen, sind unten aufgelistet. Es wird hier davon ausgegangen, daß Sie vom Heimatverzeichnis der Systemverwalterin gestartet werden (`/root`).

```
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
[Erzeuge die Dateien]
cd ..
umount /root/initrd
gzip -c -9 initrd.img > initrdgz.img
```

Es sind eine Reihe von Schritten erforderlich, die wie folgt beschrieben werden können.

1. Erzeugen Sie einen Mount-Point, ein leeres Verzeichnis, für die Initial-RAM-Disk.
2. Erzeugen Sie eine leere Datei in der benötigten Größe. Hier habe ich 1024 kB verwendet. Sie werden mehr oder weniger benötigen, je nach Verwendungszweck; die Größe ist der letzte Parameter.
3. Erzeugen Sie in der leeren Datei das Ext2-Format.
4. Mounten Sie die Datei auf den Mount-Point; dadurch wird das Loop-Device verwendet.
5. Wechseln Sie auf das gemountete Loop-Device.
6. Erstellen Sie die benötigten Dateien (siehe unten).

7. Wechseln Sie in ein anderes Verzeichnis außerhalb des Loop-Devices.
8. Entfernen Sie das Loop-Device aus der Verzeichnisstruktur (`umount`).
9. Erzeugen Sie eine komprimierte Version für die spätere Verwendung.

Inhalt der Initial-RAM-Disk

Die Dateien, die Sie auf der RAM-Disk benötigen, sind die minimalen Voraussetzungen, um irgendwelche Kommandos auszuführen zu können.

`/linuxrc`

Das Programm (Skript-Datei), das gestartet wird, um die DOS-Partition zu mounten (siehe unten).

`/lib/*`

Die Link-Programme und die Laufzeit-Bibliotheken, die von den Programmen verwendet werden.

`/etc/*`

Die Umgebung, die von den Link-Programmen verwendet wird (für alle Fälle).

`/bin/*`

Eine Shell (am besten `ash` weil sie kleiner ist als `bash`). Die Programme `mount` und `losetup` für den Zugriff auf die DOS-Partition und zum Erstellen des Loop-Devices.

`/dev/*`

Die Geräte-Dateien der verwendeten Geräte. Sie benötigen `/dev/zero` für `ld-linux.so`, `/dev/hda*` um die DOS-Partition zu mounten und `/dev/loop*` für das Loop-Device.

`/mnt`

Ein leeres Verzeichnis, um die DOS-Partition darauf zu mounten.

Die Initial-RAM-Disk die ich verwendete, ist unten aufgelistet. Der Inhalt beträgt etwa 800 kB, wenn man den zusätzlichen Platz mitberücksichtigt, der beim Abspeichern im Dateisystem benötigt wird.

```
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
```

Die einzig schwierigen Schritte betreffen die Geräte in `/dev`. Verwenden Sie das Programm `mknod`, um sie zu erstellen und gebrauchen Sie dabei die bestehenden Gerätedateien als Vorlagen, um die benötigten Parameter zu erhalten.

Die Skript-Datei /linuxrc

Die Datei `/linuxrc` auf der Initial-RAM-Disk wird benötigt, um alle Vorbereitungen zu treffen, damit das Loop-Device als Root-Partition verwendet werden kann.

Im untenstehenden Beispiel wird versucht, `/dev/hda1` als DOS-Laufwerk zu mounten und wenn das geklappt hat, die Dateien `/linux/linuxdsk.img` als `/dev/loop0` und `/linux/linuxswp.img` als `/dev/loop1` einzurichten.

```
#!/bin/sh

echo INITRD: Versuche /dev/hda1 als msdos zu mounten

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

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

else

   echo INITRD: Mounten fehlgeschlagen
   exit 1

fi
```

Das erste Loop-Device /dev/loop0 wird die Root-Partition und das zweite /dev/loop1 wird der Auslagerungsbereich (Swap-Partition).

Wenn Sie hinterher als gewöhnlicher User ohne Root-Rechte auf die DOS-Partition schreiben möchten, dann sollten Sie stattdessen das folgende Kommando verwenden:

```
mount -n -t msdos /dev/hda1 /mnt -o uid=0,gid=0,umask=000,quiet
```

Es leitet alle Zugriffe auf die DOS-Partition um und setzt die passenden Zugriffsrechte für die Dateien und Verzeichnisse.

3.4 Die Root-Partition erstellen

Die Root-Partition, die Sie verwenden werden, ist die Datei linuxdsk.img. Sie müssen sie genauso erzeugen wie die Initial-RAM-Disk. Sie muß aber größer sein. Sie können jede beliebige Linux-Distribution darauf installieren.

Der einfachste Weg wird sein, einfach eine existierende Linux-Installation dorthin zu kopieren. Die Alternative ist, ein neues Linux darauf zu installieren.

Angenommen Sie haben das getan, dann gibt es da noch ein paar kleine Änderungen, die Sie vornehmen müssen.

Die Datei /etc/fstab muß die Verweise auf die Root-Partition und die Swap-Partition unter Verwendung der beiden Loop-Devices enthalten, die mit der Initial-RAM-Disk eingerichtet wurden.

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

Das wird dafür sorgen, daß, sobald die richtige Root-Partition verwendet werden soll, diese auch vom Kernel gefunden werden kann. Außerdem kann so auch der Auslagerungsbereich auf die gleiche Weise hinzugefügt werden, wie eine Swap-Partition normalerweise verwendet wird. Sie sollten alle anderen Verweise auf eine Root- oder Swap-Partition entfernen.

Wenn Sie nach dem Start von Linux auch auf die DOS-Partition zugreifen möchten, müssen Sie noch ein paar kleine Änderungen vornehmen.

Erzeugen Sie ein Verzeichnis mit dem Namen /initrd. Dorthin wird die Initial-RAM-Disk gemountet werden, wenn das Loop-Root-Device gemountet ist.

Erstellen Sie einen symbolischen Link mit dem Namen /DOS, der auf /initrd/mnt zeigt, wohin die echte DOS-Partition gemountet wird.

Fügen Sie eine Zeile in die rc–Datei ein, die die Laufwerke mountet. Sie sollte folgendes Kommando ausführen:

```
mount -f -t msdos /dev/hda1 /initrd/mnt
```

So wird ein »falsches« Mount-Kommando auf die DOS-Partition angewendet, damit alle Programme wie z.B. df wissen, daß die DOS-Partition gemountet wurde und wo sie zu finden ist. Wenn Sie in der Datei /linuxrc andere Optionen verwendet haben, sollten Sie sie selbstverständlich auch hier verwenden.

Es ist nicht notwendig, einen Linux-Kernel auf dieser Root-Partition zu haben, weil er bereits vorher geladen wurde. Sofern Sie Kernel-Module verwenden, kopieren Sie diese wie gewohnt auf das Laufwerk.

3.5 Die Swap-Partition erstellen

Das Laufwerk, das Sie verwenden werden, ist die Datei linuxswp.img. Die Swap-Partition ist sehr einfach zu erstellen. Erzeugen Sie eine leere Datei, wie beim Erstellen der Initial-RAM-Disk und starten Sie

```
mkswap linuxswp.img
```

um sie zu initialisieren.

Die Größe des Auslagerungsbereiches ist davon abhängig, was Sie mit dem installierten System machen möchten. Ich empfehle zwischen 8 MB und der Größe des RAM-Speichers, den Sie in ihrem Rechner haben.

3.6 Das DOS-Verzeichnis erstellen

Die Dateien, die Sie verwenden wollen, müssen auf die DOS-Partition verschoben werden.

Die Dateien, die im DOS-Verzeichnis mit dem Namen `C:\LINUX` gebraucht werden, sind die folgenden:

LINUXDSK.IMG

Die Image-Datei die zur Root-Partition wird.

LINUXSWP.IMG

Der Auslagerungsbereich (Swap-Partition).

3.7 Eine Boot-Diskette erstellen

Die verwendete Diskette ist eine normale bootfähige DOS-Diskette

Unter DOS wird sie mit dem Befehl

```
format a:  /s
```

erstellt.

Auf dieser Diskette müssen Sie eine Datei mit dem Namen `AUTOEXEC.BAT` erstellen. Außerdem müssen Sie den Kernel, eine gepackte Initial-RAM-Disk und `LOADLIN.EXE` auf die Diskette kopieren.

AUTOEXEC.BAT

Die bei DOS automatisch gestartete Batch-Datei.

LOADLIN.EXE

Das Programm `LOADLIN.EXE`

ZIMAGE

Der Linux-Kernel

INITRDGZ.IMG

Das Image der gepackten Intitial-RAM-Disk.

Die Datei `AUTOEXEC.BAT` sollte nur eine einzelne Zeile enthalten:

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

Darin wird der zu ladende Kernel angeben, die Intitial-RAM-Disk und die Root-Partition, nachdem die Initial-RAM-Disk ihre Arbeit beendet hat. Außerdem wird festgelegt, daß die Root-Partition nur für lesenden Zugriff gemountet wird.

4 Das System Booten

Um von diesem neuen Laufwerk zu booten, ist es lediglich erforderlich, daß die Boot-Diskette, wie oben beschrieben vorbereitet wurde und sich in dem Computer befindet, der gestartet werden soll.

Die folgenden Vorgänge werden ablaufen:

1. DOS wird gebootet.
2. Die `AUTOEXEC.BAT` wird abgearbeitet.
3. `LOADLIN` wird gestartet.
4. Der Linux-Kernel wird in den Speicher geladen.
5. Die Initial-RAM-Disk wird in den Speicher kopiert.
6. Der Linux-Kernel wird gestartet.
7. Die Skript-Datei `/linuxrc` auf der Initial-RAM-Disk wird ausgeführt.
8. Die DOS-Partition wird gemountet und die Root- und Swap-Partition werden eingerichtet.
9. Die Boot-Sequenz wird auf dem Loop-Device fortgesetzt.

Wenn das abgeschlossen ist, können Sie die Boot-Diskette entfernen und das Linux-System verwenden.

4.1 Mögliche Probleme und Erklärungen dazu

Es gibt einige Phasen, bei denen der Startvorgang abbrechen kann. Ich will versuchen, zu erklären, welche das sind und worauf zu achten ist.

Der DOS-Boot-Vorgang ist leicht an der Startmeldung zu erkennen, die von DOS auf dem Bildschirm angezeigt wird. Wenn sie nicht erscheint, dann ist entweder die Diskette nicht bootfähig oder der PC kann nicht von dem Diskettenlaufwerk booten.

Wenn die Datei `AUTOEXEC.BAT` abgearbeitet wird, sollten die darin befindlichen Kommandos standardmäßig am Bildschirm angezeigt werden. Im vorliegenden Fall ist es lediglich die eine Zeile, die `LOADLIN` startet.

Wenn das Programm `LOADLIN` abläuft, macht es zwei leicht zu erkennende Dinge. Erstens lädt es den Kernel in den Speicher, zweitens kopiert es die RAM-Disk in den Speicher. Beides wird kommentiert durch die Meldung `Loading...`

Der Kernel beginnt sich selbst zu entpacken, dabei kann es zu CRC-Fehlern kommen, wenn das Kernel-Image fehlerhaft ist. Danach wird die Initialisierung begonnen, die viele Diagnose-Meldungen erzeugt. Das Laden der Initial-RAM-Disk ist während dieser Phase ebenfalls sichtbar.

Wenn die Skript-Datei `/linuxrc` abläuft, gibt es keine Diagnose-Meldungen, aber Sie können diese selber als Hilfe bei der Fehlersuche hinzufügen. Wenn der Schritt, das Loop-Device als Root-Partition einzurichten, fehlschlägt, dann können Sie eine Nachricht sehen, daß keine Root-Partition vorhanden ist und der Kernel bricht seine Arbeit ab.

Die normale Boot-Sequenz wird nun von der neuen Root-Partition fortgesetzt; das passiert ziemlich kommentarlos. Es könnten Probleme auftauchen, wenn die Root-Partition für schreibenden Zugriff gemountet wurde; die Option `ro` der Kommandozeile von `LOADLIN` sollte das verhindern. Weitere Probleme, die auftauchen können, sind, daß beim Boot-Vorgang die Root-Partition nicht gefunden wird. Dies wird möglicherweise durch ein Problem mit der Datei `/etc/fstab` verursacht.

Wenn der Boot-Vorgang abgeschlossen ist, bleibt das Problem, daß die Programme nicht wissen, ob eine DOS-Partition gemountet ist oder nicht. Darum ist es eine gute Idee, das »falsche« Mount-Kommando zu verwenden, das zuvor beschrieben wurde. Es macht das Leben viel einfacher, wenn Sie auf Dateien zugreifen wollen, die sich auf der DOS-Partition befinden.

4.2 Verwendete Hilfstexte, Quellenangabe

Die Texte, die ich verwendet habe, um mein erstes Loop-Root-Device zu erstellen, waren:

- Der Quell-Code des Linux-Kernels, speziell `init/main.c`.
- Die Dokumentation zum Linux-Kernel, speziell `Documentation/initrd.txt` und `Documentation/ramdisk.txt`.
- Die Dokumentation zu `LILO`.
- Die Dokumentation zu `LOADLIN`.

5 Andere Möglichkeiten des Loop-Root-Devices

Wenn einmal die prinzipiellen technischen Voraussetzungen bestehen, um ein Dateisystem zu booten, das sich in einer Datei auf einer DOS-Partition befindet, können Sie damit noch viele andere Dinge machen.

5.1 Installation zur Verwendung der DOS-Partition (ohne Boot-Diskette)

Wenn es bei Verwendung einer Start-Diskette möglich ist, Linux von einer Datei booten, die sich auf einer DOS-Festplatte befindet, dann liegt es auf der Hand, daß es auch möglich ist, dies ausschließlich unter Verwendung der Festplatte zu tun.

Es kann ein Boot-Menü verwendet werden, um `LOADLIN` aus der `AUTOEXEC.BAT` heraus zu starten. Das ergibt einen viel schnelleren Boot-Vorgang, ist ansonsten aber gleich.

5.2 Installation zum Booten mittels LILO

`LOADLIN` zu verwenden, ist nur eine Möglichkeit, um einen Linux-Kernel zu booten. Es gibt auch noch `LILO`, der so ziemlich das gleiche tut, ohne DOS zu benötigen.

In diesem Fall kann die DOS-formatierte Diskette durch eine Diskette im Ext2-Format ersetzt werden. Die sonstigen Details, mit dem Kernel und der RAM-Disk als Dateien auf der Diskette, sind sehr ähnlich.

Der Grund, warum ich die Methode mit `LOADLIN` gewählt habe, ist, daß die Argumente, die an `LILO` übergeben werden müssen, ein bißchen komplizierter sind. Außerdem ist es für einen flüchtigen Beobachter einfacher zu erkennen, welche Diskette es ist, da sie auch unter DOS gelesen werden kann.

5.3 VFAT- und NTFS-Installation

Ich habe die NTFS-Methode versucht und habe damit keine Probleme gehabt. Der Treiber für das NTFS-Dateisystem ist keine Standard-Kerneloption in der Version 2.0.x; Sie müssen den Patch von Martin von Löwis verwenden, der auf seiner Web-Seite zu finden ist:

```
http://www.informatik.hu-berlin.de/~loewis/ntfs/
```

In der Version 2.2.x ist der NTFS-Treiber standardmäßig im Kernel integriert.

Die einzigen Änderungen für die VFAT- oder NTFS-Versionen sind in der Initial-RAM-Disk. Die Skript-Datei `/linuxrc` muß ein Dateisystem des Typs VFAT oder NTFS mounten, anstatt MSDOS.

Ich wüsste keinen Grund, warum das nicht auch auf einer VFAT-Partition funktionieren sollte.

5.4 Linux installieren ohne neues Partitionieren

Das Installieren einer Standard-Distribution von Linux auf einen PC, erfordet es, von einer Diskette zu booten und die Festplatte neu zu partitionieren. Das gleiche Ziel könnte stattdessen mit einer Boot-Diskette erreicht werden, die ein leeres Loop-Device und eine Swap-Datei erzeugt. Dies würde es ermöglichen, die Installation normal fortzusetzen, aber es würde dann in ein Loop-Device installiert, anstatt auf eine eigene Partition.

Man könnte dies als Alternative zu einer Installation mittels `UMSDOS` gebrauchen. Es würde bei der Verwaltung des Festplattenplatzes effizienter sein, weil der minimale Zuordnungsbereich im Ext2-Dateisystem ein Kilobyte beträgt, anstatt bis zu 32 kB auf einer DOS-Partition. Außerdem können VFAT- oder NTFS-formatierte Laufwerke verwendet werden, die andernfalls ein Problem sind.

5.5 Booten von einem Laufwerk, das nicht bootfähig ist

Die Methode kann auch verwendet werden, um ein Linux-System von einem Laufwerk zu booten, das normalerweise nicht bootfähig ist:

- CD-ROMs
- Zip-Medien
- Parallelport-Laufwerke

Offensichtlich gibt es viele andere Laufwerke, die verwendet werden könnten. NFS-Root-Systeme sind ja bereits im Kernel als eine Option enthalten, aber die hier beschriebene Methode kann stattdessen ebenfalls verwendet werden.