The Loopback Root Filesystem HOWTO

作者: Andrew M. Bishop, *amb@gedanken.demon.co.uk*
譯者: Jing-Jong SHYUE, *shyue@sonoma.com.tw*
v1.1, 24 September 1999

這份文件解釋如何使用 Linux 的繞回設備來建立 Linux 的原生檔案系統, 使得 Linux 可以安裝在 DOS 分割區中, 並使用 Linux 原生檔案系統. 同時也討論此種技巧的其他用途.

目錄

1 簡介

1.1 版權聲明(此段不譯)

1.2 版本演變(此段不譯)

Version 1.0.0

Initial Version (June 1998)

Version 1.0.1-1.0.3

Slight Modifications, kernel version changes, typos etc. (1998 - July 1999)

Version 1.1

Added Copyright Information and Re-Submitted (September 1999)

2 繞回設備與記憶體磁碟的原理

首先我將介紹用在將繞回檔案系統設成根目錄系統所需要的基本知識.

2.1 繞回設備

繞回設備 是 Linux 中的一種虛擬設備, 用法類似於其他儲存媒體設備.

例如 `/dev/hda1`, `/dev/hda2`, `/dev/sda1` 或其他磁碟, 例如軟碟機 `/dev/fd0` 等等, 都屬於儲存設備. 這些設備都可以用來儲存檔案或是目錄結構. 也都可以依據使用需求而被格式化成各種不同的檔案系統(例如 ext2fs, msdos, ntfs 等), 並且被掛載於目錄樹中.

繞回檔案系統依存於一個儲存在別的檔案系統中的檔案, 並將這個檔案當作是一個周邊設備來操作. 這個虛擬的設備如同前面所列的眞實設備一樣, 可以被格式化或掛載於目錄樹中. 繞回檔案系統的設備檔

通常是 /dev/loop0 或是 /dev/loop1 等等, 這些設備再被指向所依存的檔案, 如此這個檔案便能被視爲虛擬設備而被掛載.

2.2 記憶體磁碟設備

在 Linux 中還有一種可以被當成檔案系統操作的虛擬設備, 也就是記憶體磁碟設備.

在這個狀況中, 這個設備僅指向部分的實體記憶體, 並不參考其他硬體設備. 這些記憶體絕不會被置換到硬碟中, 但會留存於磁碟快取中.

記憶體磁碟可以在任意時間經由寫入記憶體磁碟設備檔 /dev/ram0 或 /dev/ram1 等等, 而被產生. 記憶體磁碟同樣可以被格式化或掛載於目錄樹中, 就像繞回設備一樣.

記憶體磁碟通常被用於由軟碟啓動系統的時機, 例如 Linux 的安裝磁碟或是救援磁碟. 開機所需要的程式通常被存在一個磁碟映像檔中, 而這個映像檔通常被壓縮後存在軟碟中. 系統核心將可以認知這個檔案, 並且在開機程序中被解開後放置於記憶體磁碟中, 再被掛載成開機所需的檔案系統.

2.3 起始記憶體磁碟

在我們能將繞回設備當成根目錄系統之前, 起始記憶體磁碟是一個非常重要的機制.

因爲記憶體磁碟需要複製一個檔案系統的映像檔到記憶體中, 所以這個映像檔必須是可以被存取的. 當記憶體磁碟的工作完成要正式將眞實的根目錄系統掛載之前, 在這個記憶體磁碟中的 /linuxrc 會被執行. 而如果 /dev/initrd 存在且可用的話, 記憶體磁碟則會被掛載在 /initrd之下.

對於一個正常的開機程序而言, 這是一個相當特殊的特性, 因爲起始記憶體磁碟仍繼續執行. 而且在主要開機程序開始之前, 可以經由起始記憶體磁碟的參數來變更根目錄磁碟.

2.4 根目錄檔案系統

根目錄磁碟是第一個被掛載的裝置, 所以在開機後會成爲根目錄 "/".

由於根目錄包含了所有的檔案, 所以會有一些複雜. 在開機過程中 rc 命令稿會被執行, 這些命令稿可能是 /etc/rc.d 或 /etc/rc?.d, 端視 /etc/init 程序的設計.

在系統啓動後, 根目錄檔案系統是不可能被卸載的, 因爲所有的程式必須使用相同的資訊. 這也就是爲什麼起始記憶體磁碟是如此的重要, 因爲只有這時的根目錄系統是不同的.

2.5 Linux 開機程序

爲了了解在開機過程中起始記憶體磁碟的動作, 下列是開機的過程.

1. 核心被載入記憶體, 通常是由 LILO 或 LOADLIN 完成的. 你可以看到 Loading... 訊息, 表示正在載入核心.
2. 起始記憶體磁碟被載入, 通常也是由 LILO 或 LOADLIN 完成的. 你會再次的看到 Loading... 訊息, 表示正在載入初始記憶體磁碟.
3. 核心被初始化, 所有給核心的參數都會被設定; 同時初始記憶體磁碟被掛載爲根目錄.
4. 執行在初始記憶體磁碟上的 /linuxrc 程式.

5. 核心參數所指定的檔案系統被掛載爲根目錄.
6. 執行 /etc/init 以完成使用者所設定的開機程序.

這只是一個簡化的說明, 但足夠解釋核心的啓動與初始記憶體磁碟的用途.

3 如何建立繞回根目錄設備

下面將介紹如何建立繞回設備

3.1 需求

要建立一個繞回根目錄設備必須要有下列事項.

- 一個可以運作的 Linux 系統
- 必須能夠複製一個大檔案到 DOS 分割區中

最重要的是必須能存取一個安裝好可運作的 Linux 系統. 這是因爲繞回設備只能在 Linux 下建立. 也就是說你不可能從無中生有. 而且你必須要能在這個系統上編譯核心.

當繞回設備建立之後, 這將是一個很大的檔案. 我使用了一個 80 MB 的檔案, 如此我才能建立一個 X 終端機; 但如果你希望能做更多事, 這可能也不夠. 這個檔案必須被複製到 DOS 分割區中, 所以網路或是一堆軟碟是必須的.

你所需要的軟體包括:

- LOADLIN 1.6版以上
- 任何一個支援繞回設備的 mount 程式
- 支援所需選項的核心

對現在的 Linux 而言, 應該都能符合以上需求.

3.2 建立 Linux 核心

我是用 2.0.31 版的核心來使用繞回裝置, 其他版本的核心應該也可以用. 但他們至少必須要有以下的功能.

你必須將核心的列功能啓動

- 記憶體磁碟支援 (CONFIG_BLK_DEV_RAM).
- 初始化記憶體 (initrd) 支援 (CONFIG_BLK_DEV_INITRD).
- 繞回裝置支援 (CONFIG_BLK_DEV_LOOP).
- fat 檔案系統支援 (CONFIG_FAT_FS).

- msdos 檔案系統支援 (`CONFIG_MSDOS_FS`).

前面二個是用來使用記憶體磁碟裝置與初始化記憶體磁碟裝置. 第三個是繞回檔案系統的支援, 最後二個則是用來掛載 DOS 分割區所必須的支援.

譯核心且不使用模組支援是最簡單的選擇. 但如果你希望使用模組化支援應該也是可以, 雖然我沒有試過. 如果你使用模組化支援, 你必須確定上列的選項是直接編譯在核心內而不是編譯成模組.

依據核心版本的不同, 你可能需要使用一些修補. 下面是一些簡單的修補用來讓核心可以將繞回裝置作爲根目錄檔案系統使用.

- 2.0.0 版之前的核心: 我沒有相關資訊.
- 2.0.0 到 2.0.34 之間的核心版本: 你必須使用下列給 2.0.x 版的核心修補.
- 2.0.35 到 2.0.x 之間的核心不需要修補.
- 2.1.x 版的核心必須依據您的確實版本而使用下列給 2.0.x 或 2.2.x 版的核心修補.
- 2.2.0 到 2.2.10 之間的核心版本: 你必須使用下列給 2.2.x 版的核心修補.
- 2.3.x 版的核心: 你必須使用下列給 2.2.x 版的核心修補.

對於 2.0.x 版的核心, 必須將 `"loop", 0x0700` 這一行加到 `/init/main.c` 的原始碼中而成爲下面的樣子.

```
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 }
        };

...

}
```

對於 2.2.x 版的核心, 必須將 `"loop", 0x0700` 與前後各一行(共三行)加到 `/init/main.c` 的原始碼中而成爲下面的樣子

```
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 }
};
```

當核心組態設好後, 他必須被編譯成 zImage 檔 (make zImage). 編譯好的檔案會被放在 arch/i386/boot/zImage.

3.3 建立初始化記憶體磁碟裝置

建立初始化記憶體磁碟最簡單的方式是將它當成繞回裝置掛載. 你必須使用 root 的權限來做這件事. 建立程序如下所示, 這些指令是在 root 的家目錄 (/root) 中執行的.

```
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
[建立必須的檔案]
cd ..
umount /root/initrd
gzip -c -9 initrd.img > initrdgz.img
```

這些指令的作用如下所述.

1. 建立一個給初始化記憶體使用的掛載點 (一個空目錄).
2. 建立一個指定大小的空檔案, 此處我建立一個 1024kB 的檔案, 你必須依據你所要放置檔案的量來決定這個檔案的大小. (檔案大小是最後一個參數)
3. 將這個檔案格式化成 ext2 檔案系統.
4. 將這個檔案以繞回的方式掛載倒掛載點上.
5. 切換目錄到繞回設備的掛載點中.
6. 建立必須的檔案 (細節將在後面討論).
7. 離開繞回檔案所掛載的目錄.

8. 將繞回設備卸載.
9. 建立一個壓縮的版本以供後續使用.

初始化記憶體磁碟的內容

你必須至少有下列的檔案放置於初始化記憶體磁碟中, 否則將不能使用任何指令.

- /linuxrc 掛載 DOS 檔案系統的指令稿. (如下所示)
- /lib/* 程式所需要的動態連結器與函式庫.
- /etc/* 動態連結所需的快取, 雖然靜態連結程式不需要, 但可以避免警告訊息.
- /bin/* 使用者命令解譯器 (此處我使用 ash, 因爲它比 bash 小). 而 mount 與 losetup 程式是用於掛載 DOS 磁碟與設定繞回裝置.
- /dev/* 必須的裝置控制檔, 你至少需要 /dev/zero 以供 ld-linux.so 使用, /dev/hda* 以掛載 DOS 磁碟, 以及 /dev/loop* 以掛載繞回裝置.
- /mnt 一個空的掛載點, 以供 DOS 磁碟掛載.

下面是我所用的起始化記憶體磁碟之內容, 所需空間約 800 kB.

```
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
```

比較複雜的步驟是建立 dev 目錄下面的裝置控制檔. 你可以用 /dev 下面現有的裝置控制檔爲範本, 並以 mknod 程式來建立他們.

/linuxrc 指令檔

起始化記憶磁碟中的 /linuxrc 指令稿是用來完成掛載繞回裝置所需的前置準備.

下面的範例是以 /dev/hda1 爲 DOS 分割區, 並在成功掛載 DOS 分割區後, 將/linux/linuxdsk.img 設定爲 /dev/loop0 ;並將 /linux/linuxswp.img 設定爲 /dev/loop1.

```
#!/bin/sh

echo INITRD: Trying to mount /dev/hda1 as msdos

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: Mount failed
   exit 1

fi
```

前面的 /dev/loop0 裝置將被掛載爲根目錄, 後面的 /dev/loop1 則將被作爲記憶體置換區使用.

如果你希望 root 以外的使用者也能寫入 DOS 分顆區, 你必須改成使用 mount -n -t msdos /dev/hda1 /mnt -o uid=0,gid=0,umask=000,quiet. 這會將所有對 DOS 分割區的存取對應成 root, 並且適當的設定權限.

3.4 建立根目錄裝置

你所使用的根目錄裝置將是 `linuxdsk.img` 這個檔案. 你可以用與建立起始記憶體磁碟相同的方法建立這個檔案, 所不同處僅在於必須較大, 足以容納你要安裝的程式.

最簡單的辦法是將現有的 Linux 系統複製到這個檔案系統中. 當然你也可以重新安裝一套新的 Linux 系統.

假設你已經做過這些事, 你只要作一些小改變.

`/etc/fstab` 檔案必須將根目錄磁碟與置換區對應到你在啓示記憶體磁碟中的繞回設備.

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

這將確保核心會使用正確的根目錄裝置. 也可以用相同的方法加入置換磁碟. 你也必須移除其他根目錄磁碟與置換區的對應.

如果你希望在 Linux 啓動後能夠讀取 DOS 分割區, 你必須作下面這些額外的小修改.

建立叫做 `/initrd` 的目錄, 這是再繞回根目錄系統掛載後, 初始記憶體磁碟掛載的地方.

建立一個符號連接 `/DOS` 指向 `/initrd/mnt`, 也就是 DOS 分割區被掛載的地方.

在 rc 檔中加入類似 `mount -f -t msdos /dev/hda1 /initrd/mnt` 的指令, 以建立一個 "假的" 掛載, 如此例如 `df` 等程式才會知道這個掛載. 如果你在 `/linuxrc` 中用不同的設定, 請自行修正以符合設定.

由於核心在之前便會被載入, 所以你不需要在根目錄裝置中放置系統核心. 但如果你使用了核心的模組, 你仍然必須將這些模組放在根目錄裝置中.

3.5 建立置換裝置

你所使用的置換裝置將是 `linuxswap.img`. 這個置換裝置相當容易建立. 首先如同之前的做法建立一個空檔案, 然後執行 `mkswap linuxswap.img` 來格式化這個置換檔.

置換空間的大小依據個人的使用而不同, 但我建議將它設爲大於 8MB, 但小於實體記憶體量.

3.6 建立 MSDOS 目錄

你所要使用的檔案必須被搬移到 DOS 分割區中.

必須被放在 `C:\LINUX` 目錄下的檔案包括

- `LINUXDSK.IMG` 作爲根目錄裝置的影像檔.
- `LINUXSWP.IMG` 置換裝置.

3.7 建立開機軟碟

開機軟碟的製作與標準 DOS 開機軟碟相同.

在 DOS 下使用 `format a: /s` 以建立開機軟碟.

在這張軟碟上, 你必須建立一個 AUTOEXEC.BAT 檔, 如下所示. 並且複製 Linux 系統核心與壓縮過的起始記憶體磁碟檔案, 以及 LOADLIN 執行檔.

- AUTOEXEC.BAT - DOS 下的開機自動執行批次檔.
- LOADLIN.EXE - LOADLIN 執行檔.
- ZIMAGE - Linux 系統核心.
- INITRDGZ.IMG - 壓縮過的起始記憶體磁碟檔.

AUTOEXEC.BAT 僅需要下面這行.

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

這會指定要用的系統核心與起始化記憶體磁碟檔以及根目錄磁碟的位置, 並將根目錄以唯讀的方式掛載.

4 啓動系統

要用這個根目錄裝置啓動必須將前面所做的開機軟碟放於 PC 中, 並由這個軟碟開機.
你將會看到下列的程序

1. DOS 啓動
2. AUTOEXEC.BAT 執行
3. LOADLIN 執行
4. Linux 核心被複製到記憶體中
5. 起始記憶體磁碟被複製到記憶體中
6. Linux 核心開始執行
7. 在起始記憶體磁碟中的 /linuxrc 命令稿被執行
8. DOS 分割區被掛載, 且根目錄裝置與置換裝置被設定
9. 由繞回裝置完成開機程序

開機完成後, 你可以取出開機軟碟並使用這個 Linux 系統.

4.1 可能發生的問題與解決方式

開機程序可能在許多關卡上失敗, 我將試著解釋他們與該如何檢查.

DOS 的啓動很容易辨別, 他應該會顯示 MS-DOS Starting .... 如果沒有顯示, 則這個軟碟可能是不可開機的, 或電腦不是由這台軟碟機開機.

當 AUTOEXEC.BAT 被執行的時候, 所有被執行的指令預設會顯示在螢幕上. 在此處應該只有一行, 並以 LOADLIN 開頭.

當 LOADLIN 執行時, 應該有二樣很容易看的的事情發生. 首先是他會將核心載入記憶體中. 接著會將起始記憶體磁碟載入記憶體. 這二個都可以由 Loading... 訊息確認.

核心將自己解壓縮, 而且當核心檔案損壞時, 將顯示 **crc** 錯誤. 當核心開始執行時, 會顯示一堆初始化程序與檢查訊息. 初始記憶體磁碟的載入也會在此階段顯示.

當 /linuxrc 開始執行時, 不會有可供診斷的訊息產生, 但你可以自己加入輔助訊息來檢查. 在這個程序中, 如果在將繞回裝置當成根目錄裝置時失敗的話, 你將會得到 "沒有根目錄裝置" 的訊息, 且核心會放棄執行.

接下來定義在根目錄裝置中的正常開機程序會被啓動. 你有可能得到根目錄被掛載成可讀寫的警告, 但在 LOADLIN 指令中的 'ro' 可以解決這個問題. 當開機程序弄不清哪個是根目錄裝置時, 會產生一些問題, 這個問題多半來自 /etc/fstab 中的錯誤.

當開機程序完成後, 剩下的問題是程式搞不清楚 DOS 分割區被掛載在何處. 這也是爲何要執行一個"假"的掛載指令, 這將使你較容易存取 DOS 分割區.

4.2 參考文件

在我建立第一個入回根目錄檔案系統時, 我參考了下列文件:

- Linux 核心原始碼, 特別是 init/main.c
- Linux 核心說明文件, 特別是 Documentation/initrd.txt 以及 Documentation/ramdisk.txt.
- LILO 說明文件.
- LOADLIN 說明文件.

5 其他繞回根目錄裝置的可能用途

當由 DOS 分割區啓動 Linux 的方法確立之後, 我們將可以用在下面的用途上.

5.1 將 Linux 安裝在僅有 DOS 分割區的硬碟上

既然可以利用軟碟與存在 DOS 磁碟中的一個檔案來啓動 Linux 系統, 那也可以完全使用 DOS 磁碟來啓動系統.

一個設定好的開機選單可以用來從 AUTOEXEC.BAT 中啓動 LOADLIN, 這能使開機速度較快, 但卻會與其他部分相關.

5.2 利用 LILO 啓動

使用 LOADLIN 只是其中一種啓動 Linux 核心的放式. 另外可以使用 LILO 來啓動且不需要使用 DOS.

在這個狀況下 DOS 格式的軟碟被換成 ext2fs 格式的軟碟. 其他細節都非常相似, 必須將系統核心與初始記憶體磁碟放於該軟碟上.

我選擇使用 LOADLIN 的原因在於 LILO 需要較複雜的指令. 而且可以在 DOS 下使用的軟碟也較容易處理.

5.3 將 Linux 安裝在 VFAT / NTFS 檔案系統上

我曾經試過將 Linux 安裝在 NTFS 之上, 而且沒也任何問題. 但在標準的 2.0.x 版核心中, NTFS 並不被支援, 但你可以在 *http://www.informatik.hu-berlin.de/˜loewis/ntfs/* 找到修補檔. 在 2.2.x 版的核心中, NTFS 已被支援.

使用在 VFAT 或 NTFS 時與本文範例不同之處僅在於製作起始記憶體磁碟時, 編寫 `/linuxrc` 指令搞以掛載檔案系統時, 必須以 vfat 或 ntfs 爲參數而非 msdos.

據我所知沒有理由不能用在 VFAT 分割區上.
(譯註: 弟是用在 VFAT 上面, 並以 100 MB ZIP 啓動 Linux)

5.4 安裝 Linux 而不需重新格式化

正常安裝 Linux 套件時都必須由軟碟機開機, 並重新格式化硬碟. 利用本文的方法也可以建立繞回檔案, 並繼續正常的安裝程序, 並將系統安裝於繞回裝置而不是一個分割區中.

這可以作爲安裝在 `UMSDOS` 外的另一種選擇, 這將會更有效率的使用磁碟空間, 因爲在 ext2fs 中最小配置單位是 1kB, 而在 DOS 中則是 32kB. 同時也可以用在 VFAT 或 NTFS 上而不會產生其他問題.

5.5 從不可開機的設備上啓動 Linux

這個方法也可以用在正常不能用來開機的設備上啓動 Linux 系統.

- 光碟片
- Zip 磁碟片
- 平行埠磁碟裝置

還有許多其他的裝置可以使用, NFS 根目錄系統已經被核心所支援, 所以本文的方法應該也可以用.