無磁碟 Linux Mini Howto

作者: Robert Nemkin buci@math.klte.hu
譯者: 謝崑中 HSIEH Kun-Chung, 台灣 中華民國 Taiwan, Republic of China (NOT People's Republic of China) (*linuxer.bbs@cis.nctu.edu.tw*)
v0.0.3版, 1996年 9月 12日

本文件在敘述如何設立一個無磁碟 Linux 機器。著作權爲 Robert Nemkin 所有。著作權條款依照 GPL。我想要謝謝 Bela Kis <bkis@cartan.math.klte.hu> 將這份文件翻譯成爲英文。

目錄

1 修改

- v0.0.3版 1996年9月12日: 一小部份的修改

2 如何設立一個無磁碟的 Linux 機

本文件在敘述如何設立一個無磁碟 Linux 機器。有時候我們可能需要在一個既無硬碟 亦無軟碟機的個人電腦上執行 Linux。如果有個網路， 其他的 Unix 系統有著 bootp, tftp, NFS 伺服器，而我們又有一個 eprom 燒錄機，那麼我們有辦法在建立一個無任何磁碟機的 Linux系統了。

3 相關文章

- NFS-root Mini Howto
- Linux NET-2/3-HOWTO 由 Terry Dawson 所寫, `94004531@postoffice.csu.edu.au`
- `/usr/src/linux/README` 關於建構以及編譯新核心

4 硬體

在本文所提到的都是以下面的建構來做的

- Sun-OS 4.1.3 做爲啓動伺服器
- Slackware 2.3 + Linux 1.2.8 + wd 8013 網路卡
- 已使用中的乙太網路

5 基本概念

基本概念如下：個人電腦會經由 bootp 協定從啓動伺服器中來取得它的 IP 位址。 用 0.0.0.0 爲初始 IP 位址，而它的核心程式會經由 tftp 協定。[1]

以下是我們的步驟

5.1 設定個人電腦

取得 nfsboot 套件 (可從你較喜歡的 linux mirror 鏡射站的 `/pub/Linux/system/Linux-boot` 目錄取得) 。它包含了一個給 wd8013卡可直接燒錄用的 booteprom 映像檔。

還有其他方法可以準備個人電腦部份:

- 如果你的機器並不是沒有磁碟的，那麼你可以使用小小的 DOS 程式或
- 含有同樣套件的二進位軟體映像檔。

[1]跳過區段的啓動 (經由路由器router) 並不是個簡單問題，所以，不是將伺服器以及無磁碟機的機器放置在同一個網路 區段，就只有建構一個 UDP 輔助位址在你的 router 到伺服器的位址上。細節請參考你的 router 產器手冊。

如果你選擇的是第二者，你必須使用 dd 命令來將這個映像檔寫入一張軟碟片裏。

這個映像檔包含了 bootp 以及 tftp 客户端程式。 你也需要準備一個 linux 核心程式碼，它含有 nfs-root 選項。

- 如果你使用最新的穩定核心程式，如 linux-1.2.13，那麼你需要使用包含在 nfsbootk套件內的 patch 修補核心程式。[2]
- 如果你想要用最新的，但卻不是很穩的 linux-1.3.x 系列核心，那麼你就需要建構 nfs-root 這選項。

隨便你要不要建構區塊設備 (軟碟機或硬碟機) 的支援，但你必須建構 tcp/ip 支援、 wd 乙太網路卡支援、 nfs 檔案系統支援。然後重新編譯核心程式。

5.2 在伺服器上設定 bootpd

它可以在 bootpd-2.4.tar.gz套件中找到 (可從你較喜歡的 linux mirror 鏡射站的 /pub/Linux/system/Network/boot.net 目錄內)。取得這個套件， 編譯並安裝它。如果你其他的 Unix 機器也是用 Slackware Linux 的，那麼標準的供應版本就 有 bootpd 了，所以你可以跳過這部份。 daemon 可以直接發出命令來執行

```
        bootpd -s
```

或是使用 inetd，但你需修改：

- /etc/inetd.conf 在這些條列中移除 # 號

```
     # tftp   dgram   udp     wait    root    /usr/sbin/in.tftpd      tftpd /export
     # bootps dgram   udp     wait    root    /usr/sbin/in.bootpd     bootpd
```

- 在 /etc/services 中插入或是移除備註符號：

```
     bootps          67/tcp          # BOOTP server
     tftp            69/udp          # TFTP server
```

- 重新啓動 inetd，使用

```
          kill -HUP <process id of inetd>.
```

5.3 在伺服器上建構 bootpd

由先， bootpd 有個叫 bootptab 的建構檔，它通常被放在 /etc 內。你必須修改 它，加上你的無磁碟機器的閘道 IP 位址、 dns 伺服器，以及乙太網路位址(es)。 範例如下 /etc/bootptab:

```
  global.prof:\
          :sm=255.255.255.0:\
          :ds=192.168.1.5:\
          :gw=192.168.1.19:\
          :ht=ethernet:\
          :bf=linux:
  machine1:hd=/export/root/machine1:tc=global.prof:ha=0000c0863d7a:ip=192.168.1.140:
  machine2:hd=/export/root/machine2:tc=global.prof:ha=0800110244e1:ip=192.168.1.141:
  machine3:hd=/export/root/machine3:tc=global.prof:ha=0800110244de:ip=192.168.1.142:
```

[2]參考 patch(1)

global.prof 是 host 項目的一般樣本，其中

- sm 欄包含子網路遮罩
- ds 欄包含網域名稱伺服器的位址
- gw 欄包含内定閘道位址 gateway
- ht 欄包含網路媒介硬體種類
- bf 欄包含啓動檔檔名

之後，每一台機器應該有下面這行

- 首欄包含主機名稱
- hd 欄包含啓動檔案的目錄
- 全區樣本應該被包含到 tc 欄，
- ha 欄包含乙太網路卡的硬體位址
- ip 欄包含被分配的的 IP 位址

5.4 瞭解 tftp

TFTP (`Trivial File Transfer Protocol`) 是一個檔案傳輸協定，就像 ftp 一樣，但 它比比較簡化，因此可以容易地在 EPROM 裏來編碼。 TFTP 的使用方法有二種：

- 簡單的 tftp: 表示客户端可以存取你的整個檔案系統。雖然較簡單但卻開了一個安全大洞 (任何人都可以經由 tftp 來拿到你的密碼)。
- 安全的 tftp: tftp 伺服器會使用chroot.2 系統呼叫來改變它自己的根目錄。任何在新目錄以 外的事件都完全無法存取。因爲 chroot 目錄變成新的根目錄， 硬碟在 bootptab 整檔時必須反應這個新的處境。例如：當我們使用不安全的 tftp 時， hd 欄包含了啓動目錄的完整路徑，：`/export/root/machine1`。 當使用安全的 tftp 時， /export 被當成根目錄，而 /export 變成 /，且 hd 欄就變成 `/root/machine1`了。

幾乎每一個 Unix 實作都包含 tfpt 伺服器，也許你並不需要自己安裝一個。

5.5 在遠端伺服器上設定一個小型的 Linux 建構

這包含 Slackware供應版内的 a, ap, n, 及 x 部份。要安裝更多的套件也可以；但 是上面的套件已足夠一個無磁碟 X 終端機的需求了。首先你需要一個已在運轉的 Linux 系統。在遠端機器裏找一個磁碟空間並 export 它成爲可讀寫並掛上它到 Linux 機器的檔案系統上的某一處 (例如 /mnt)。開始 Linux 設定並改變root 選項從 / 變爲 /mnt。然後設立如上面所說的套件，如果你只是要執行一個無磁碟的 Linux 那 麼就不需要做其他改變。另一方面，如果你計畫使用超過一個以上的無磁碟機器， 那麼上面的設定就需要些許的修改，因爲有些檔案以及目錄必須只有機器本身使用。 我們可以以移開 /usr 目錄(它包含非私人的資料)的方式跳過這個問題， 然後建立一個分開的子目錄給每一個無磁碟機器。例如，如果 /export/linux/machine1 被掛在 /mnt 上，那麼在初始設定後目錄結構會像這樣

```
/export/linux/machine1/bin
/export/linux/machine1/sbin
/export/linux/machine1/lib
/export/linux/machine1/etc
/export/linux/machine1/var
/export/linux/machine1/usr
```

改變後你會有

```
/export/linux/machine1/bin
/export/linux/machine1/sbin
/export/linux/machine1/lib
/export/linux/machine1/etc
/export/linux/machine1/var
/export/linux/usr
```

現在為其他的機器建立目錄。從現在開始我們假設你的無磁碟機器分別叫做 machine1、machine2、machine3等等；使用以下的 bash 命令手稿來設立其他的目錄

```
cd /export/linux
for x in machine2 machine3 ; do
        mkdir $x; cd $x
        (cd ../machine1; tar cf - *) | tar xvf -
done
```

然後做下列的 export:

- /export/linux/usr 唯讀（對每個使用者）
- /export/liunx/machine1 只有對 machine1 有 rw,root 權利。
- /export/liunx/machine2 只有對 machine2 有 rw,root 權利。
- /export/liunx/machine3 只有對 machine3 有 rw,root 權利。

如下[3]:

```
# This file is /etc/export
# for remote linux X terminals by Buci
# this line is only once
/export/root/usr              -access=linuxnet
# these lines once for every host
/export/root/machine1       rw=machine1,root=machine1
/export/root/machine2       rw=machine2,root=machine2
/export/root/machine3       rw=machine3,root=machine3
```

請不要忘了執行 exportfs -a

[3]下面範列是依照 SunOs 4.1.3 的 exports 檔案的語法來做

5.6 建構 tftp 伺服器

現在該是建構 tftp 伺服器的時候了。如果你不需要一個安全的 tftp 那很簡單， 因爲你的客户端已經可以從 /export 目錄來啓動。

如果一個安全的 tftp 被採用，那麼你可以在 /tftpboot 目錄下做個完整的 /export/linux 目錄結構。 (用一個實體核心程式及連結給其他機器的符號連結)， 或是讓 /export 目錄成爲安全的 tftpd 的根目錄。或者如果你有一另外的 tftpboot 目錄，那麼，同樣地，你只需要原來的目錄結構，然後用一個實體核心程式來 符號連結其他機器)。 你可以執行下列命令來完成這個設定：

```
mkdir -p /tftpboot/export/linux/machine1
cd /tftpboot/export/linux/machine1
cp /export/linux/machine1/<name of the kernel> .
```

然後執行下面：

```
mkdir -p /tftpboot/export/linux/machine2
cd ../machine2
ln -s ../machine2/<name of the kernel>
```

5.7 最後工作

最後，你必須插入

```
/sbin/mount nfs_server:/export/linux/usr /usr
```

到

```
/export/linux/<machinex>/etc/rc.d/rc.S
```

的第一行。在這裏 <machinex> 指的是 machine1, machine2等等。

6 記憶體以及磁碟空間要求; 速度

我只在 Slackware 2.3上測試過;對於其他的供應商(Linux)套件或版本，下面的數字可能會不同。

- 磁碟空間: 28MB + 6.5MB/每台機器
- RAM: 我在 8MB 上跑 X，只需要 4MB swap。我猜想，這個可以被建立一爲每一部機器分別做一在 /tmp裏面，請不要忘了執行 mkswap。
- 速度: 我在一台有 8 Megs 的 486 DX2/66 機器上跑並無問題。

7 可能的錯誤

- 我發現一個奇怪的錯誤：在 /dev 子目錄裏， SunOS 搞錯了設備名稱，所以我被迫 將這個子目錄掛上一個有磁碟的 Linux 機器上來重新執行 MAKEDEV。 (原因是因爲 linux nfs 和 SunOs nfs是不同的： 兩者都使用32 位元在主要及次要設備名稱， 但是linux 使用 16 位元寬度欄在兩者上，SunOs 則使用 14 位元寬度欄給主要設備， 18 位元寬給次要設備號碼。)

- 當無磁碟 linux 被啟動時，只有一個行徑 route 會被放在 tftp 伺服器的行徑表裏 , 所以你需要設定正確的行徑表。你有兩種選擇：
  - 手動建構每一台機器的每一個 rc.S。
  - 或使用bootp 客户端套件並寫一個通用的設定命令手稿

8 本文件中的錯誤以及更詳細的解說

- 正確引用相關的文件。
- SunOs 是以 BSD 爲基礎，所以需要包含 SVR4 (如 Solaris) 爲基礎的伺服器建構。
- 雖然 Linux 和 SunOs 都可以做爲 bootp/tftp 伺服器，但是 Linux 會比較有用些。
- 更新這份文件到最新的 etherboot 套件。
- 指出核心程式版本1.2.13 在nfs root 修補後與新的 1.3.x 版(此版已包含了 nfs-root 的修補)的不同處。
- 需要試試 wd8013 以外的其他乙太網路卡
- 包含 bootpc的建構資訊，一個bootp 客户端來設定正確的 rooting 表。
- 編排上或其他的錯誤：麻煩您告訴我一聲，謝謝。 `buci@math.klte.hu`

譯者按: 若您發現本文件內容有翻譯不順暢或錯誤的地方，請您來信告訴我，以做更正，謝謝。 *linuxer.bbs@cis.nctu.edu.tw*。