Linux以太網-HOWTO

by Paul Gortmaker　　v2.8, Oct 29, 2000

本文為以太網-HOWTO，匯集了有關哪些以太網裝置可以用于Linux，以及如何進行設置的信息。請注意本文側重于以太網卡的硬件與底層驅動方面的內容，并未涵蓋諸如`ifconfig`和`route`等軟件方面的問題。對于那些問題，請參閱Network-HOWTO。

目錄

1 導言

以太網-HOWTO包括了該買什么網卡與不該買什么網卡；如何設置網卡，怎樣運行多個網卡，以及其它常見錯誤和問題。它包含了對目前所有能夠得到的最常見網卡的支持的細節資料。

它不包含有關軟件方面的內容，因為NET-3 Howto中包含了這些內容。還要注意這里不包括（或者至少不應當包括）那些非Linux特有的有關以太網的一般問題的解答。對于那些問題，可以參考*comp.dcom.lans.ethernet*的FAQ中的大量資料，與所有其它新聞組FAQ一樣通過FTP到`rtfm.mit.edu`獲取。

本修訂版覆蓋了Linux 2.2.17及其之前的內核發行版本。

以太網-HOWTO的作者為:

Paul Gortmaker, `p_gortmaker@yahoo.com`

原始的ASCII版本的以太網-HOWTO的主要資料來源為：

Donald J. Becker, `becker@scyld.com`

感謝他編寫了許多Linux目前使用的以太網卡驅動程序。他也是最初的NFS服務器的作者。謝謝Donald！

1.1 本文檔的新版本

本文檔的新版本可以從如下主頁獲得：

Ethernet-HOWTO <http://metalab.unc.edu/mdw/HOWTO/Ethernet-HOWTO.html>

望采用FTP方式或者希望獲取非HTML格式文檔的可使用以下站點：

Sunsite HOWTO Archive <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/>

這是“官方”站點——也可以在其它各種Linux WWW/ftp鏡像站點找到本文檔。在有了新資料或新驅動程序時本文檔將進行更新。如果你閱讀的文檔已經存在6個月以上了，那么你需要檢查一下是否已經有更新過的版本。

本文檔有多種格式（postscript，dvi，ASCII，HTML，等等）。推荐用HTML格式瀏覽（通過WWW瀏覽器）或Postscript/dvi格式瀏覽。這兩種格式都包含有普通文本ASCII格式無法提供的交叉引用功能。

1.2 以太網-HOWTO的用法

由于本指南越來越大，你可能不希望花一個下午從頭讀到尾。而令人高興的是你不必閱讀全部內容。HTML和Postscript/dvi版本都有內容列表，可以幫助你更快地找到所需的內容。

可能你是由于無法使網卡工作，而且不知道該怎么做或者如何檢查，才來閱讀本文檔的。下一節(1.3 (幫幫我——網卡沒反應！))就是針對Linux初學者并給出正確的指導。

經常有一些問題被不同的人問了一遍又一遍。可能你碰到的某個問題就是常見問題之一，而且在本文檔的FAQ部分(2 (常見問題))已經有了答案。每個人在發文尋求幫助之前都最好先看一下這一節。

如果你還沒有網卡，那么你可能想從決定買網卡開始。(2.1 (該為Linux買什么網卡？))

如果你已經有了一塊以太網卡，但不能確定是否可以在Linux上使用它，那么你可能想讀一下包含每一個制造廠家及其網卡的特定信息的章節。(4 (有關銷售商/制造廠家/型號的信息))

如果你對Linux設備驅動程序的一些技術細節感興趣，那么你可以瀏覽有此類信息的章節。(7 (技術信息))

1.3 幫幫我——網卡沒反應！

好，別緊張。下面就告訴你如何解決問題，即使你此前對Linux或以太網硬件一無所知。

首先你需要弄清楚你的網卡類型，然后確定Linux是否有此類型網卡的驅動程序。主機在控制不同的網卡時使用的方法一般也不一樣，而Linux驅動程序（如果有的話）就以允許Linux使用此卡的格式包含了這樣的控制信息。如果你沒有手冊一類的東西讓你知道網卡的類型，那么你可以嘗試處理“神秘”網卡的章節(參考章節：4.44 (識別未知網卡))。

現在你知道了你的網卡類型，閱讀特定網卡章節中有關你的網卡的細節（參考章節：4 (有關銷售商/制造廠家/型號的信息)），那里以字母順序列出了網卡制造廠家、各自的類型號以及是否有Linux驅動程序。如果在列表中被表明“不支持”，你就可以放棄了。如果從中找不到你的網卡，那么看一下你的網卡手冊是否說它與某個已知網卡類型“兼容”。例如至少有几百種網卡與原來的Novell NE2000設計兼容。

假設你已經發現了你的網卡有Linux驅動程序，現在就找到并利用它。因為Linux有你的網卡的驅動程序并不意味著它存在每一個內核中。（內核是啓動時最先載入的操作系統核心，包含各種硬件的驅動程序，以及其它等等。）可能只有很少几個預先生成的內核，和一大批較小的作為分立模塊的驅動程

序，或者是許多的內核，覆蓋了大量內建驅動程序的組合，這都完全取決于是誰生成了你所使用的Linux發行版本。

現在的大多數Linux發行版本都提供大量較小的作為分立模塊的不同驅動程序。需要的模塊通常可以在啓動過程的后期載入，或者在存取特定設備時以命令的形式載入一個驅動程序。你需要在內核啓動后把該模塊加載到內核上。查閱你所用的發行版本中有關安裝和使用模塊的信息，并參考本文檔的有關模塊章節。(9.2 (把以太網驅動程序作為模塊使用))

如果你沒有找到包含你的網卡驅動程序的已生成內核，也沒有找到該驅動程序的模塊形式，那么可能是你的網卡不夠常見，而且需要你自己生成一個包含該驅動程序的內核。只要你已經安裝了Linux，生成一個定制的內核一點兒都不難。你只要對內核需要包含些什么回答是或者不是，然后讓它生成就行了。有一個Kernel-HowTo可以幫助你。

此時你還需要以某種方式啓動內建了你的驅動程序的內核，或者以模塊的形式載入驅動程序。大約人們碰到的問題有一半兒出在沒有以這種或那種方式載入驅動程序，所以你會發現網卡現在工作了。

如果它還不工作，那么你需要証實一下內核確實檢測到了網卡。要做到這一點，你需要在系統啓動以及所有模塊載入之后鍵入`dmesg | more`。這可以讓你查看啓動過程中內核所顯示的啓動信息。如果網卡被檢測到，你可以從中找到以`eth0`開頭的網卡驅動程序顯示信息，指出網卡所使用的驅動程序名稱和硬件參數（中斷設置、I/O地址等等）。（注意：在啓動時，Linux會列出所有系統中安裝的PCI卡，無論是否有驅動程序——不要把這個誤認為其后的驅動程序檢測！）

如果你沒看到這樣的驅動程序識別信息，那么驅動程序就沒有檢測到你的網卡，這就是問題所在。查看FAQ(2 (常見問題))尋找網卡未檢測到時該做些什么。如果你的網卡是NE2000兼容的，在FAQ中有一些有關NE2000網卡檢測的技巧。

如果網卡被檢測到了，但檢測信息報告了一些錯誤，比如資源沖突，那么驅動程序可能無法正常初始化，網卡還是不可用。這一類的最常見錯誤信息也在FAQ中列出來了，并給出了相應的解決方案。

如果檢測信息看起來沒問題，那么再檢查一下驅動程序報告的網卡資源是否與網卡的物理設置一致（網卡的物理設置或者是通過網卡上的跳線，或者是網卡制造廠家提供的軟件工具完成。）它們必須完全一致。例如，網卡的跳線或配置為IRQ 15，而驅動程序在啓動信息中報告為IRQ 10，那么就會出問題。在FAQ中討論了最常見的驅動程序錯誤地檢測各種網卡配置信息的情況。

現在你已經使你的網卡以正確的參數被檢測到了，但愿一切正常。如果還有問題，那么或者是軟件配置有錯，或者是硬件配置有錯。軟件配置錯誤就是在`ifconfig`和`route`命令中沒有設置正確的網絡地址，其處理細節可以在Network HowTo和“Network Administrator's Guide”中找到，它們可能都在你用來安裝的CD-ROM上。

硬件配置錯誤是某些資源沖突或忘記配置（即在啓動時沒檢測到驅動程序），使網卡無法正常工作。通常可以用几種不同的方法來檢查。(1) 在`ifconfig`試圖打開設備來使用時報告出錯信息，如“SIOCSFFLAGS: Try again”。(2) 驅動程序報告`eth0`出錯信息（用`dmesg | more`查看）或者每次試圖發送或接收數據時奇怪的不一致。(3) 鍵入`cat /proc/net/dev`來顯示`eth0`的errs、drop、fifo、frame或carrier列中的非零數字。(4) 鍵入`cat /proc/interrupts`顯示網卡的零中斷計數。大多數典型的硬件配置錯誤也在FAQ中進行了討論。

好吧，如果你現在還有麻煩，閱讀本文檔的FAQ，閱讀有關銷售商章節中有關你的網卡的細節，如果問題還沒解決，那么你只好求助于某個相應的新聞組了。在求助文章中請給出所有相關的細節資料，比如網卡商標、內核版本、驅動程序的啓動信息、`cat /proc/net/dev`的輸出、清晰的問題描述，當然還有你在試圖解決問題時嘗試采用的方法。

你可能會奇怪有那么多的人發出諸如“Can someone help me? My ethernet doesn't work.”而沒有其它內容的無用信件。新聞組的讀者一般會忽略掉這些愚蠢的信件，而詳細的有內容的問題描述則會讓“Linux專家”立即指出你的問題所在。當然在用電子郵件發送問題報告時這也同樣成立——總是提供盡

可能多的信息。

1.4 網卡應該支持的電纜類型

使用RJ-45（大的電話插頭）連接器的雙絞線從技術上來說叫做10BaseT。你可能聽到過它被稱為UTP（無屏蔽雙絞線）。

使用BNC（金屬的推進和旋轉鎖定）連接器的以太網細纜（RG-58同軸電纜）從技術上來說叫做10Base2。

只會在早期的裝置上發現的老式以太網粗纜（10mm同軸電纜）被稱為10Base5。在某些以太網卡上使用的15針D形插座（AUI連接器）是用來連接以太網粗纜和外部收發器的。

大多數以太網卡都有價格稍高$10-$20的"Combo"版本。它們同時提供雙絞線和細纜以太網收發器，可以讓你以后改變主意。

大多數裝置使用10BaseT /100BaseT，因為10Base2無法升級為100Base之類的網絡。10Base2對于因為某種原因不想購買HUB的愛好者建立室內網絡而言是足夠的。

參考5 (電纜、同軸電纜......)以了解有關以太網線纜不同類型的情況。

2 常見問題

這里是一些有關使用Linux進行以太網連接的常見問題。某些特定問題按照制造商進行分類。可能你想問的問題別人已經問過（而且被回答了！），所以即使沒有在這里找到你的答案，還可能在諸如*Dejanews* `<http://www.dejanews.com>`之類的新聞檔案中找到你所要的。

2.1 該為Linux買什么網卡？

這個問題的答案很大程度取決于你希望用網絡連接干些什么，以及會遇到多大的數據流量。

如果你只想作為一個用戶偶爾使用一下FTP或WWW，那么即使是8比特的老ISA網卡都能滿足你的需要。

如果你想建立一個服務器，并要求接收或發送網絡數據給CPU帶來的負載最小，那么你可能需要看一下使用具有總線控制能力的芯片的PCI網卡，比如DEC tulip (21xxx)芯片，或者AMD PCnet-PCI芯片。

如果你的需求在這二者之間，那么使用有穩定的驅動程序的便宜的PCI網卡或16比特的ISA網卡就可以了。

2.2 Alpha驅動程序——獲取與使用

我聽說我的網卡有一個更新的或初步的/alpha驅動程序。從哪兒得到它呢？

最新的"新"驅動程序可以在Donald的FTP站點：`cesdis.gsfc.nasa.gov`里面的`/pub/linux/`下找到。因為事情變化很頻繁，可能需要四處找一找。或者使用WWW瀏覽器去：

Don's Linux Home Page `<http://cesdis.gsfc.nasa.gov/linux/>`

查找你想要的驅動程序更簡單一些。（留神WWW瀏覽器會悄悄地把源碼中的TABs替換為空格，等等——如果無法確定的話，使用FTP下載，至少也得用WWW瀏覽器的FTP URL。）

如果驅動程序確實是alpha版本，或pre-alpha版本，那么請恰當地對待它。換句話說，不要抱怨，因為你無法弄清用它能做些什么。如果弄不清如何安裝，最好不要去試。同樣，如果它使你的機器宕機了，不要抱怨。相反，你應該發給我們一份材料組織很好的Bug報告，如果是一個補丁，那就更好！

注意，某些“可用”的實驗性/alpha驅動程序已經包含在標准的內核源碼樹中。在運行`make config`時你首先要回答的一個問題就是“Prompt for development and/or incomplete code/drivers”。在此你要回答“Y”以包括任何alpha/實驗性驅動程序。

2.3 一台機器使用多個以太網卡

做些什么才能讓Linux運行兩塊以太網卡？

這個問題的答案取決于驅動程序是否被用做可載入的模塊或者直接編譯進了內核。大多數Linux發行版本現在都使用模塊化的驅動程序。這樣就不用發行許多內核，每種內核設置一個不同的內建驅動程序。使用一個單一的基本內核，如果特定用戶系統需要，一旦系統啓動，就可以從驅動程序模塊文件（通常存放在`/lib/modules/`）中載入個別的驅動程序。

把驅動程序作為模塊使用：對于PCI驅動程序，模塊通常會自動檢測該品牌類型所有安裝的網卡。但對于ISA網卡，探尋一個網卡是不安全的操作，因此你需要提供網卡的I/O地址以便模塊知道去哪里查找。這一信息存儲在文件`/etc/conf.modules`中。

例如，如果一個用戶有兩塊ISA NE2000網卡，一塊在`0x300`，一塊在`0x240`，它們在`/etc/conf.modules`文件中顯示如下：

```
alias eth0 ne
alias eth1 ne
options ne io=0x240,0x300
```

這几行的意義：就是說如果管理員（或內核）進行`modprobe eth0`或者`modprobe eth1`，那么為`eth0`或者`eth1`載入`ne.o`驅動程序。此外，在載入`ne.o`模塊時，使用選項`io=0x240,0x300`，這樣驅動程序就知道去哪里尋找網卡。注意`0x`很重要——DOS里常用的`300h`在這里沒有用。改變`0x240`和`0x300`的順序會使哪一塊物理網卡以`eth0`和`eth1`結尾發生改變。

同這個例子一樣，大多數ISA模塊驅動程序可以接受多個以逗號分隔的I/O值以處理多塊網卡。但是，某些（老的？）驅動程序，比如3c501.o模塊，目前載入一個模塊只能處理一塊網卡。這樣，要檢測兩塊網卡就必須載入兩次該模塊。此時，文件`/etc/conf.modules`將如下所示：

```
alias eth0 3c501
alias eth1 3c501
options eth0 -o 3c501-0 io=0x280 irq=5
options eth1 -o 3c501-1 io=0x300 irq=7
```

在此例中，選項`-o`用來給每個模塊實例一個唯一的名字，因為不能用同一個名字載入兩個模塊。選項`irq=`也是用來指定網卡設置的硬件IRQ。（此方法也能用于可接受多個以逗號分隔的I/O值的模塊，但這樣會使模塊被不必要地載入兩次，降低了效率。）

最后一個例子，假設用戶有一塊在`0x350`的3c503網卡和一塊在`0x280`的SMC Elite16 (wd8013)網卡。則應該這樣：

```
alias eth0 wd
```

```
alias eth1 3c503
options wd io=0x280
options 3c503 io=0x350
```

對于PCI網卡，只要用alias語句把ethN接口和相應的驅動程序名聯系起來就行了，因為PCI網卡的I/O地址可以被安全地檢測到。

可用的模塊一般存放在/lib/modules/‘uname -r‘/net下，這里uname -r命令可以得到內核的版本（比如2.0.34）。你可以在這里看看哪一個驅動程序適合你的網卡。一旦你在conf.modules文件里進行了正確的設置，就可以用下面的方法檢查一下：

```
modprobe ethN
dmesg | tail
```

這里“N”是你要檢測的以太網卡的接口號。

使用編譯進內核的驅動程序：如果你需要的驅動程序編譯進了內核，那么處理多塊以太網卡的接口已經存在了。但缺省情況下只自動檢測一塊以太網卡。這樣就避免了啓動時檢測敏感網卡可能引起的麻煩。

（注意：在2.1.x之后的內核中，啓動檢測被分為安全和不安全的兩類，所有安全的檢測（如對PCI和EISA網卡）可以自動找到所有相關的網卡。在至少有一塊ISA網卡的多網卡系統中還需要進行以下的處理。）

有兩種方法可以啓動對第二塊（或第三塊等等）網卡的自動檢測。最簡單的方法是向內核傳遞啓動參數，由LILO完成。使用ether=0,0,eth1這樣簡單的啓動參數就可以完成對第二塊網卡的檢測。此時按照啓動時找到的網卡順序分配eth0和eth1。假如你想讓0x300處的網卡為eth0，而0x280處的網卡為eth1，那么可以使用

```
LILO: linux ether=5,0x300,eth0 ether=15,0x280,eth1
```

命令ether=可以接受的參數并不僅限于如上所示的IRQ + I/O + name。請參看9.1 (傳遞以太網參數......)以了解全部的句法、網卡特定參數和LILO使用技巧。

這些啓動參數可以固定，這樣就不用每次都必須重新敲一遍。參看LILO手冊中有關LILO的配置選項“append”。

第二種方法（不建議使用）是編輯文件Space.c并用零替換I/O地址中的0xffe0入口。0xffe0入口是用來告訴內核不要檢測該設備——把它替換為零就啓動了對該設備的自動檢測。

2.4 這個以太東東不干活。為什么？

如上所述，ether=命令只對編譯進了內核的驅動程序起作用。現在大多數的發行版本都用模塊的方式使用驅動程序，所以很少再使用ether=命令了。（某些早期文檔需要更新以反映這一變化。）如果你想使用模塊化的以太網驅動程序的選項，必須修改/etc/conf.modules文件。

如果你是使用編譯的驅動程序，而且已經把ether=加進了LILO配置文件，需要重新運行lilo使更新后的配置文件生效。

2.5 NE1000/NE2000網卡（及其兼容卡）的問題

問題： 在用v2.0.x啓動時沒有檢測到PCI NE2000兼容網卡。

原因： 在v2.0.30之前的`ne.c`驅動程序只知道基于RealTek 8029的兼容網卡的PCI ID號。在此只后才出現了使用其它PCI ID號的PCI NE2000兼容網卡，所以驅動程序無法檢測這些網卡。

解決方案： 最簡單的方法是把Linux內核升級到v2.0.31以上版本。它可以識別五種不同的NE2000-PCI芯片的ID號，在啓動或載入模塊時自動檢測到它們。如果你升級到了2.0.34以上版本，會有一個比原先的ISA/PCI驅動程序稍小但更高效的PCI專用NE2000驅動程序。

問題： 啓動時PCI NE2000兼容網卡被報告為ne1000（8比特網卡！）或者在v2.0.x下載入ne.o模塊不起作用。

原因： 某些PCI兼容網卡不支持字節存取（因此不是百分之百兼容的NE2000）。這使它在檢測時被誤認為NE1000網卡。

解決方案： 你需要升級到v2.0.31以上版本。現在的驅動程序會檢測到這種硬件Bug。

問題： PCI NE2000網卡的性能很差，即使按照性能技巧章節所説的減小了窗口大小。

原因： 十多年前設計和出售的初始8390芯片的技術數據手冊上提到，為了得到最高的可靠性，在每次寫操作之前需要一個讀操作。驅動程序能夠輕易地做到這一點，但從v1.2內核時代起，缺省情況下取消了這一操作。有一個用戶報告説重新啓用這一“錯誤的特性”就可以改善廉價的PCI NE2000兼容網卡的性能。

解決方案： 由于只有一個用戶提出報告把它作為解決方案，不要對此寄予太大的希望。重新使用寫之前的讀操作可以簡單地編輯`linux/drivers/net/`下的驅動程序文件，取消包含`NE_RW_BUGFIX`的那一行的注釋，然后重建內核或載入相應的模塊。如果這樣確實有效，請給我發一封e-mail，描述性能上的差異和你所使用的網卡/芯片類型。（對`ne2k-pci.c`驅動程序也可以如法炮制。）

問題： `ne2k-pci.c`驅動程序對PCI NE2000網卡報告諸如`timeout waiting for Tx RDC`錯誤信息，無法正常工作。

原因： 你的網卡或網卡到PCI總線的連接無法處理該驅動程序使用的長字I/O優化。

解決方案： 首先，檢查BIOS/CMOS設置，看看與PCI總線相關的計時對于可靠的操作是否過于嚴格了。否則，使用ISA/PCI的`ne.c`驅動程序（或者刪除`ne2k-pci.c`中的`#define USE_LONGIO`），使你的網卡可用。

問題： 沒檢測到ISA的即插即用NE2000網卡（如RealTek 8019）。

原因： 初始的NE2000特性不支持即插即用，因此Linux的NE2000驅動程序也不支持即插即用。

解決方案： 使用網卡所附的DOS配置盤取消PnP，并為該網卡設置一個指定的I/O地址和IRQ。在`/etc/conf.modules`里增加這樣的一行`options ne io=0xNNN`，其中`0xNNN`是你為網卡設置的16進制I/O地址。（假設你使用的是模塊化驅動程序；否則，在啓動時使用一個`ether=0,0xNNN,eth0`參數）。你也可以進入BIOS/CMOS設置，把IRQ從PnP改為Legacy-ISA。如果你需要為兼容其它的操作系統而保留PnP設置，那么你可以看一下*isapnptools*軟件包。使用`man isapnp`看看它是否已經安裝在你的系統上了。如果沒有，瀏覽一下下面的連接：

ISA PNP Tools `<http://www.roestock.demon.co.uk/isapnptools/>`

問題： 在啓動檢測時NE*000驅動程序報告“not found (no reset ack)”。

原因： 這跟上面所説的改動有關。在証實8390處于所檢測的I/O地址之后，進行重新設置。在網卡完成重新設置后，應當通知重新設置完成。你的網卡沒有通知，所以驅動程序認為不存在NE網卡。

解決方案： 你可以在啓動時使用一個未被使用的`mem_end`的16進制值`0xbad`，告訴驅動程序你有一個壞網卡。在使用`0xbad`覆蓋時你必須為網卡提供一個非零的I/O地址。例如，在`0x340`的網卡不響應重新設置時，則使用如下方法：

```
LILO: linux ether=0,0x340,0,0xbad,eth0
```

這樣，即使你的網卡不響應重新設置，網卡檢測還能繼續下去。如果你是以模塊方式使用驅動程序，那么可以象提供I/O地址一樣提供選項`bad=0xbad`。

問題： NE*000網卡使機器在第一次網絡訪問時死機。

原因： 這個問題從早期的1.1.57內核到現在都出現過，而且只在有限的几種軟件配置的兼容網卡上出現。看來是需要用某些特殊的方法來初始化它們。

解決方案： 有几個人報告在熱啓動（即loadlin或Ctrl-Alt-Del）Linux之前，運行提供的DOS軟件配置程序或提供的DOS驅動程序可以使網卡工作。這表明這些卡需要以某種特殊的方式初始化，與當前的Linux驅動程序稍有區別。

問題： 在`0x360`的NE*000以太網卡沒有檢測到。

原因： 你的NE2000網卡在I/O空間占據了`0x20`個字節，使它與`0x378`處的并口發生沖突。其它可能的設備是`0x370`處的第二個軟盤控制器（如果有的話），以及`0x376-0x377`處的第二個IDE控制器。如果該口已被其它驅動程序注冊，內核將不再進行檢測。

解決方案： 或者把你的網卡移到`0x280`，`0x340`，`0x320`一類的地址，或者在編譯時不支持并行打印機。

問題： 每次打印時網絡都斷開（NE2000）。

原因： 與上一個問題相同，但你的內核比較老，不支持對重疊I/O區域的檢查。使用如上的解決方案，有空時獲取一個新的內核。

問題： 沒檢測到0xNNN: 00 00 C5 ... 處的NE*000以太網卡。（非法標識yy zz）

原因： 首先，你是否在地址0xNNN處有一個NE1000或NE2000網卡？如果有，報告的硬件地址是否象一個合法地址？如果是的話，那么你的NE*000兼容網卡很差勁。所有的NE*000兼容網卡都假定網卡上的SA PROM的第14和15字節為`0x57`。而你的網卡不是這樣——它的值為“yy zz”。

解決方案： 有兩種解決方法。最簡單的方法就是如上所述的“no reset ack”解決方案，使用一個`0xbad`的mem_end值。這樣在提供一個非零的I/O地址時就可以忽略標識檢查。此方法無需重新編譯內核。

第二種方法（對黑客）需要修改驅動程序，并重新編譯內核（或模塊）。在驅動程序（/usr/src/linux/drivers/net/ne.c）的42行有一個“Hall of Shame”列表。這個表是用來檢測那些差勁的兼容網卡的。例如，DFI網卡在PROM的前三個字節使用“DFI”，而不是象要求的那樣在第14和15字節使用`0x57`。

問題： 機器在啓動時出現“8390...”或“WD....”信息后死機。拔掉NE2000就好了。

原因： 你的NE2000兼容網卡兼容性不好。一個激活的NE2000是個無底洞，會使其它的驅動程序陷在其空間內進行自動檢測。把NE2000改到一個不常用的地址就可以從其它的自動檢測中消除這一陷阱，機器也就可以啓動了。

解決方案： 把你的NE2000地址改為`0x340`一類的地址。此外，你可以在和“ether=”參數一起使用“reserve=”啓動參數，保護網卡不受其它設備驅動程序檢測的影響。

問題： 機器啓動時在進行SCSI檢測時死機。

原因： 這個問題跟上面的一樣，改變以太網卡的地址，或使用reserve/ether啓動參數。

問題： 機器啓動時在進行聲卡檢測時死機。

因： 不對，實際上是發生在靜默方式的SCSI檢測過程中，與上面的問題一樣。

問題： 啓動時檢測不到NE2000 – 根本就沒有啓動信息。

解決方案： 因為造成檢測不到的原因很多，所以沒有“神奇的解決方案”。下面列出了可能有所幫助的一些措施。

1) 構建一個只包含需要的設備驅動程序的內核。証實你確實是從新內核啓動的。忘記運行lilo等會使你從老的內核啓動。（仔細觀察啓動時報告的構建時間/日期。）聽起來很明顯，但我們以前都犯過這個錯。通過檢查`System.map`文件里`ne_probe`一類的名稱，確定驅動程序已包含在新的內核里。

2) 仔細觀察啓動信息。看看它是否提及正在進行諸如“NE*000 probe at 0xNNN: not found (blah blah)”一類的ne2k檢測，或者就是靜悄悄地失敗了。這里的區別很大。使用`dmesg|more`在登錄后瀏覽啓動信息，或者在啓動完成顯示登錄提示符時使用Shift-PgUp卷回屏幕。

3) 啓動后，執行`cat /proc/ioports`，証實網卡要求的全部I/O空間是空的。如果網卡在`0x300`，那么ne2k驅動程序要求的空間為`0x300-0x31f`。如果其它設備的驅動程序注冊了其中的一個口，就不會對該地址進行檢測，而是靜悄悄地檢測下一個要檢測的地址。常見的情況是lp驅動程序保留了

```
0x378
```

，或者第二個IDE通道保留了`0x376`，這就使ne驅動程序停止檢測`0x360-0x380`。

4) 與上面一樣執行`cat /proc/interrupts`。確定沒有其它設備注冊了你為以太網卡設置的中斷。這種情況下，檢測可以進行，以太網卡驅動程序會在啓動時大聲抱怨無法得到所要求的IRQ中斷線。

5) 如果你還為驅動程序靜悄悄地失敗而苦惱，那么編輯并給檢測增加一些printk()。比如，對于ne2k，你可以在`linux/drivers/net/ne.c`中增加/刪除某些行（用“+”或“-”表示），如下所示：

```
    int reg0 = inb_p(ioaddr);

+    printk("NE2k probe - now checking %x\n",ioaddr);
-    if (reg0 == 0xFF)
+    if (reg0 == 0xFF) {
+       printk("NE2k probe - got 0xFF (vacant I/O port)\n");
        return ENODEV;
+    }
```

那么它就會輸出檢查的每一個口地址信息，你可以看到你的網卡地址是否被檢測了。

6) 你還可以從Don的ftp站點（在howto中也提及了）獲取ne2k的診斷程序，看看你在啓動進入Linux后能否用它檢測你的網卡。使用“`-p 0xNNN`”選項告訴它在哪里尋找你的網卡。（缺省情況下只檢測`0x300`，與啓動時的探測不同，不會檢測其它的地址。）在找到網卡時的輸出如下：

```
Checking the ethercard at 0x300.
  Register 0x0d (0x30d) is 00
  Passed initial NE2000 probe, value 00.
8390 registers: 0a 00 00 00 63 00 00 01 00 30 01 00 00 00 00
```

```
SA PROM  0: 00 00 00 00 c0 c0 b0 b0 05 05 65 65 05 05 20 20
SA PROM 0x10: 00 00 07 07 0d 0d 01 01 14 14 02 02 57 57 57 57

        NE2000 found at 0x300, using start page 0x40 and end page 0x80.
```

你的注冊值和PROM值可能會不一樣。注意，對16比特網卡，所有PROM值都增加一倍，以太網卡地址（00:00:c0:b0:05:65）出現在第一行，加倍后的`0x57`標識出現在PROM的結尾。

在`0x300`處沒有安裝網卡時的輸出如下：

```
Checking the ethercard at 0x300.
  Register 0x0d (0x30d) is ff
  Failed initial NE2000 probe, value ff.
8390 registers: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
SA PROM       0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
SA PROM 0x10: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

 Invalid signature found, wordlength 2.
```

出現值`0xff`的原因是在讀取空I/O口時返回的就是該值。如果在檢測的區域內有其它硬件，你可以看到一些非`0xff`的值。

7) 嘗試在運行提供的DOS驅動程序或配置程序之后，從DOS啓動軟盤（通過loadlin）熱啓動進入Linux。這可能會進行一些額外的（即非標准的）“魔法”來初始化網卡。

8) 試一下Russ Nelson的ne2000.com包驅動程序，看它能否看見你的網卡——如果還不行，事情就不大妙了。例如：

```
A:> ne2000 0x60 10 0x300
```

所用參數為軟件中斷向量、硬件IRQ和I/O地址。你可以從任意的msdos檔案文件中的pktdrv11.zip里找到它——現在的版本大概是11以上了。

2.6 SMC Ultra/EtherEZ和WD80*3網卡的問題

問題： 你得到了如下信息：

```
eth0: bogus packet size: 65531, status=0xff, nxpg=0xff
```

原因： 是共享內存的問題。

解決方案： 最普遍的原因是配置的PCI機器沒有映射到ISA內存設備里。因此你讀到的是PC的RAM（全都是`0xff`值），而不是存放接收數據包數據的網卡上的RAM。

另一個容易解決的典型問題是板卡沖突，在此區域有緩存或“shadow ROM”，或者你的ISA總線運行速度高于8Mhz。以太網卡上的內存失效的數目也令人驚奇，所以如果你的以太網卡有診斷程序的話，運行一下。

問題： SMC EtherEZ在非共享內存（PIO）模式下不工作。

原因： 老版本的Ultra驅動程序只支持共享內存模式下的操作。

解決方案： 版本2.0以上的內核所附驅動程序就支持可編程I/O模式的操作。升級到v2.0以上版本。

問題： 老的wd8003或可跳線的wd8013總是得到錯誤的IRQ。

因： 老的wd8003網卡或可跳線的wd8013兼容卡沒有驅動程序可以從中讀取設置的IRQ的EEPROM。如果驅動程序無法讀到IRQ，就嘗試用auto-IRQ發現它。若auto-IRQ返回0，那么驅動程序就給8比特網卡分配IRQ 5，或者為16比特網卡分配IRQ 16。

解決方案： 使auto-IRQ代碼無效，并在你的模塊配置文件（對于內建的驅動程序則通過啓動參數）告訴內核你把網卡跳成了什么IRQ。

問題： SMC Ultra網卡被檢測成了wd8013，但IRQ和共享內存地址是錯的。

原因： Ultra網卡看起來跟wd8013很相象，如果內核里沒有Ultra驅動程序，wd驅動程序就會把ultra誤認為wd8013。ultra檢測在wd之前，所以一般不會出問題。ultra在EEPROM保存的IRQ和內存地址與wd8013保存的位置不同，所以報告的值是假的。

解決方案： 只保留需要的驅動程序重新編譯內核。如果你在同一台機器上同時使用wd和ultra網卡，并使用模塊，那么首先載入ultra模塊就行了。

2.7 3Com網卡的問題

問題： 3c503選擇了IRQ N，但其它設備也需要IRQ N。（比如CD ROM驅動程序、modem等。）可以不編譯進內核就解決這個問題嗎？

解決方案： 3c503驅動程序按照順序{5, 9/2, 3, 4}檢測空閑的IRQ線，從中找到一個未被使用的IRQ。在網卡被`ifconfig`操作配置時選擇中斷IRQ。

如果你使用的是模塊化的驅動程序，可以用模塊參數設置各種情況，包括中斷IRQ的值。

下面的語句選擇IRQ9、基址`0x300`、<ignored value>和if_port #1（外部收發器）。

```
io=0x300 irq=9 xcvr=1
```

另外，如果驅動程序被編譯進了內核，你還可以通過LILO在啓動時傳遞參數來設置同樣的值。

```
LILO: linux ether=9,0x300,0,1,eth0
```

下面的語句選擇IRQ3、檢測基址、<ignored value>和缺省if_port #0（外部收發器）。

```
LILO: linux ether=3,0,0,0,eth0
```

問題： 3c503: configured interrupt X invalid, will use autoIRQ.

原因： 3c503網卡只能使用中斷IRQ{5, 2/9, 3, 4}中的一個（這些是網卡所能連接的中斷線。）如果你使用一個不在其中的IRQ值，就會得到如上的提示。一般情況下，沒必要為3c503指定中斷值。3c503會在ifconfig配置時使用autoIRQ，并從IRQ{5, 2/9, 3, 4}中選擇一個。

解決方案： 使用上述的合法IRQ值，或者不指定IRQ以啓用autoIRQ。

問題： 提供的3c503驅動程序無法使用AUI（粗纜以太網）端口。怎樣才能不使用缺省的細纜以太網端口而選擇AUI端口？

解決方案： 3c503的AUI端口對于內建驅動程序可以在啓動時選擇，對于模塊化驅動程序可以在插入模塊時選擇。這一選擇會覆蓋未使用的dev->rmem_start變量的低比特位，所以啓動參數：

```
LILO: linux ether=0,0,0,1,eth0
```

可以對內建在內核的驅動程序起作用。

要在載入模塊時指定AUI端口，只需把`xcvr=1`附加在模塊選項包含你的I/O和IRQ值的那一行就行了。

2.8 非特定網卡的FAQ。

2.8.1 Linux與ISA的即插即用以太網卡。

要獲得最佳效果（問題最少），推荐使用隨網卡附的程序（通常是DOS程序）取消PnP機制，并給網卡設置一個固定的I/O地址和IRQ。確定使用的I/O地址在啓動時被驅動程序檢測到，如果使用模塊，則在`/etc/conf.modules`中使用`io=`選項提供地址。也可以進入BIOS/CMOS設置，把IRQ從PnP改為Legacy-ISA（如果計算機有此選項的話）。

注意，運行基于DOS的配置程序一般并不需要安裝DOS。可以用DOS軟盤啓動，然后從提供的軟盤上運行它們就可以了。同樣可以自由地下載OpenDOS和FreeDOS。

如果需要使用PnP以與其它操作系統兼容，就得每次啓動時都使用Linux的isapnptools包配置網卡。還需要確定為網卡選擇的I/O地址被驅動程序檢測到，或用`io=`選項提供I/O地址。

有些系統在BIOS/CMOS設置菜單中有“enable PnP OS”（或類似的名稱）選項，几乎在所有情況下或網卡無法正常工作甚至檢測不到的情況下都需要禁用此選項。一個用戶對此描述得很好“我不知道它在屏幕后干了些什么，但看起來是在搗亂。”

2.8.2 啓動時沒有檢測到以太網卡。

出現這個問題的常見原因是人們使用的內核不支持特定的網卡。對于模塊化的內核，這一般說明要求的模塊尚未被載入，或者需要用模塊選項指定其I/O地址。

如果使用的是模塊化的內核，就象大多數用Linux發行版安裝的那樣，試著用一下該發行版的配置工具來選擇網卡所用模塊。對于ISA網卡一個較好的主意是，確定網卡的I/O地址，如果配置工具要求選項則把它作為一個選項（如`io=0x340`）加進去。如果沒有配置工具，那么需要在`/etc/conf.modules`里添加正確的模塊名稱（及選項）——閱讀`man modprobe`以了解更多的細節。

如果使用的是發行版套件里的預編譯內核，那么需要查看文檔以確定安裝的是哪一種內核，以及是否支持所用的網卡。如果不支持的話，要么試著找一個支持所用網卡的內核，要么自己生成一個內核。

只保留所需的驅動程序生成自己的內核是個聰明的主意，因為這會減小內核大小（為應用程序保留寶貴的RAM！），減少打擾敏感硬件的設備檢測數目。生成內核并不象聽起來那么復雜。你只需要對一些有關想要哪些驅動程序的問題回答是或不是，其它的事都由程序完成。

另一個主要原因是其它的設備占用了網卡所需的部分I/O空間。大多數網卡在I/O空間里占用了16或32個字節。如果你的網卡設在`0x300`并需要32個字節，那么驅動程序就要求`0x300-0x31f`。如果某個其它設備驅動程序注冊了哪怕其中一個端口，驅動程序就不會對該地址進行檢測，而是靜悄悄地進入下一個檢測地址。所以，在啓動之后，運行一下`cat /proc/ioports`以確定網卡要求的全部I/O空間都是空的。

還有一個問題就是網卡跳到的I/O地址不是缺省檢測的地址。每個驅動程序的檢測地址列表可以很容易地在驅動程序源碼中的文本注釋里找到。即使網卡設定的I/O地址不在檢測地址列表上，你也可以

在啓動時用`ether=`命令提供（對內建驅動程序），參見9.1 (傳遞以太網參數...)。模塊化的驅動程序可以在`/etc/conf.modules`里使用`io=`選項指定一個非缺省檢測的地址。

2.8.3 `ifconfig`報告了錯誤的網卡I/O地址。

這不可能。你只是理解錯誤。這不是一個Bug，而且報告的數字是正確的。這只出現在某些基于8390的網卡上（如wd80x3、smc-ultra等），實際的8390芯片位于第一個給定I/O端口加上一個偏移量處。此偏移量保存在`dev->base_addr`里，也就是`ifconfig`報告的值。如果你想看到網卡使用的全部端口，試一下`cat /proc/ioports`以得到想要的數字。

2.8.4 PCI機器探測到了網卡，但驅動程序檢測失敗。

某些PCI的BIOS在上電時沒有啓用所有的PCI卡，特別是在使用了"PNP OS"的BIOS選項情況下。這一特性是為了支持當前依然使用某些實模式驅動程序的Windows版本。或者禁用該選項，或者升級到一個可以啓用被禁用網卡的新驅動程序。

2.8.5 PCI機器里的共享內存ISA網卡不工作（`0xffff`）

這常表現為顯示讀出大量`0xffff`值。除非你正確地設置了PCI ROM BIOS/CMOS SETUP配置，任何類型的共享內存網卡都不會在PCI機器上工作。你必須把網卡所用內存區域設置為可以從ISA總線訪問共享內存。如果你不知道哪些設置有用，那么詢問你的供應商或者當地的計算機大拿。對于AMI BIOS，在"Plug and Play"部分有一個"ISA Shared Memory Size"和"ISA Shared Memory Base"的設置。對于類似wd8013和SMC Ultra的網卡，把共享內存的大小從缺省的禁用改為16kB，并把基址改為網卡的共享內存地址。

2.8.6 網卡看來在發送數據，但沒有收到過數據。

執行`cat /proc/interrupts`。這樣產生的列表會顯示網卡產生的所有中斷事件的實時數目。如果為0或在試圖使用網卡時沒有增加，那么可能是與計算機安裝的其它設備發生物理中斷沖突（無論其它的設備是否安裝/提供了驅動程序）。把其中一個設備的IRQ改成未使用的IRQ。

2.8.7 異步傳輸模式(ATM)支持

Werner Almesberger在進行Linux的ATM的支持工作。他使用的是Efficient Networks的ENI155p板（*Efficient Networks* `<http://www.efficient.com/>`）和Zeitnet的ZN1221板（*Zeitnet* `<http://www.zeitnet.com/>`）。

Werner説ENI155p的驅動程序已經很穩定了，而ZN1221的驅動程序目前還沒有完成。

去下面的連接查看一下最新的進展：

Linux ATM Support `<http://lrcwww.epfl.ch/linux-atm/>`

2.8.8 吉比特以太網支持

Linux支持吉比特以太網嗎？

是的，目前至少已經有了兩個驅動程序。在v2.0和v2.2內核里有一個Packet Engines G-NIC PCI吉比特以太網適配器的驅動程序。驅動程序的更多細節、支持和更新可訪問：

`http://www.scyld.com/linux/drivers/yellowfin.html`

v2.2內核提供的`acenic.c`驅動程序可用于Alteon的AceNIC吉比特以太網卡和其它如3Com的3c985一類的基于Tigon的網卡。這個驅動程序還可以用于NetGear的GA620，但還需要証實。

2.8.9 FDDI支持

Linux支持FDDI嗎？

是的。Larry Stefani為v2.0編寫了Digital的DEFEA（FDDI EISA）和DEFPA（FDDI PCI）網卡驅動程序。它被包含進v2.0.24內核。目前還沒有其它的網卡被支持。

2.8.10 全雙工支持

全雙工能達到20MBps嗎？Linux支持嗎？

Cameron Spitzer對全雙工10Base-T網卡有如下論斷："如果你連在全雙工交換HUB上，你的系統足夠快而且不做太多其它的工作，它會使你的網絡在兩個方向上都保持忙碌。不存在什么全雙工的10BASE-2或10BASE-5（細纜和粗纜）。全雙工是通過取消適配器的碰撞檢測來達到的。這就是為什么用同軸電纜實現不了全雙工；LAN無法以全雙工方式運轉。10BASE-T（RJ45接口）使用不同的線進行發送和接收，所以二者可能同時進行。交換HUB處理碰撞問題。信號速率是10Mbps。"

所以，你只能以10Mbps速率接收或發送數據，無法期望得到兩倍的性能提高。對于是否支持，取決于網卡和可能的驅動程序。有些網卡可以自動協商，有些需要驅動程序支持，還有的需要用戶在網卡的EEPROM配置中設置選項。只有那些認真的用戶會注意到全雙工與半雙工模式間的差別。

2.8.11 SMP機器上的Linux以太網卡

如果有錢買多處理器（MP）的計算機，那么最好買一個好點兒的以太網卡。對v2.0內核這還不是個問題，但對v2.2就成問題了。大多數老式的非智能（如ISA總線的PIO和共享內存設計）網卡在設計時根本沒考慮多處理器應用。簡單地說就是買一個現代設計的智能網卡，并確定有能夠處理多處理器操作的驅動程序。（注意這里的"現代設計" - PCI-NE2000就是在現代總線上有10多年歷史的老式設計。）在驅動程序的源碼里查找`spin_lock`可以很好地說明該驅動程序是否能夠處理多處理器操作。下面詳細解釋了為何要為多處理器應用購買好的網卡（以及不買會出現什么問題）。

在v2.0內核，在任意時刻只有一個處理器允許進入"內核態"（即改變內核數據或運行設備驅動程序）。所以從網卡（及相關驅動程序）的角度來看，這與單處理器操作沒有什么不同，所以不會出問題。(這也是得到一個可以工作的Linux多處理器版本的最簡單的辦法——使用一個大鎖使得一次只有一個處理器處于內核狀態。這樣你就知道不可能有兩個處理器同時要修改同一數據！）

在任意時刻只有一個處理器允許進入內核態的不利之處在于只有運行自我控制和密集計算的程序時才會獲得多處理器的優越性。如果程序進行了大量諸如向磁盤或網絡讀/寫數據的I/O操作，在處于內核的那個處理器努力運行所有的設備驅動程序以滿足I/O請求的同時，其它的處理器都必須等待自己的I/O請求被處理完成。這樣內核就成為了瓶頸，由于只有一個處理器運行在內核態，多處理器機器的性能在I/O任務重、單鎖的情況下迅速降級到接近單處理器的水平。

很明顯這與理想情況相差太遠（尤其是對于文件/WWW服務器、路由器等），v2.2的內核就使用了粒度更小的鎖——也就是說同時可以有多個處理器進入內核。不再是對整個內核使用一個大鎖，而是使

用許多較小的鎖保護關鍵數據，防止同時被多個處理器控制——例如，一個處理器可以運行網卡驅動程序，同時另一個處理器可以運行磁盤驅動器的驅動程序。

好的，這樣就有問題了：更小的鎖定就意味著可以有一個處理器試圖通過以太網驅動程序發送數據，同時另一個處理器試圖訪問同一個驅動程序/網卡來做別的事情（比如通過`cat /proc/net/dev`得到網卡統計數據）。哎呦——你的網卡正在通過網線發數據，你又要用它來收數據。網卡被同時要求做兩件事（或更多），會弄糊涂的，所以有可能在處理過程中網卡使你的機器死機。

因此，為單處理器寫的驅動程序不再適用——它需要更新控制對網卡訪問的鎖，使得網卡的接收、發送和操作配置數據這三種任務以網卡穩定操作所要求的程度串行化起來。沒有更新為使用穩定多處理器操作的鎖的驅動程序在輕的網絡負載下可能看起來會正常工作，但在兩個（或更多）處理器試圖同時進行多個任務時就會造成死機，或至少表現出奇怪的行為，這就是問題。

更新后的意識到多處理器的以太網驅動程序將要求一個驅動程序范圍的鎖，使得內核進入驅動程序的訪問入口被限制為一次一個。這樣，任務就被串行化，而對硬件的處理就如同在單處理器下一樣，也就一定應當穩定。使用驅動程序范圍的鎖的不利之處在于它類似于對整個內核加鎖（但規模較小）對性能的影響——也就是說，一次只可以有一個處理器處理網卡。 [技術提示：如果增加的鎖是`irqsave`類型的而且被持有較長時間，對性能的影響還包括增加了中斷延遲。]

這里可以進行的改進有兩處。可以嘗試減少獲得與釋放鎖之間所用的時間，或者在驅動程序內部實現更為細化的鎖（比如滿足網卡需求的前提下，把整個驅動程序的鎖替換為若干保護同時訪問若干敏感寄存器或設置的鎖）。

但是，對于老式的非智能網卡而言，在設計時根本就沒有考慮過多處理器的應用，這樣的改進可能無法實現。更糟的是非智能網卡一般要求處理器在網卡和內存之間傳送數據，所以在最壞的情況下，每當在ISA總線傳送1.5kB數據包時，鎖都被一直保持著。

現代的智能網卡一般無需處理器的幫助就可以直接在網卡和內存之間傳遞網絡數據。這是個很大的改進，因為只需要在處理器通知網卡使用哪一塊內存保存下一個網絡數據包的那一小段時間持有鎖。現代的網卡在設計時同樣也不要求對整個驅動程序使用一個大鎖。

2.8.12 Alpha/AXP PCI板上的Linux以太網卡

對于v2.0，只有3c509、depca、de4x5、pcnet32和所有8390驅動程序（wd、smc-ultra、ne、3c503等等）是編寫成“結構無關”的，所以它們可以運行在基于DEC的Alpha CPU系統上。其它一些從Donald的WWW主頁上下載的更新過的PCI驅動程序也可以工作，因為它們也是按照結構無關的思想編寫的。

注意，使驅動程序與結構無關所需要進行的改動并不很復雜。只需要如下進行：

——把所有與`jiffies`有關的值都乘以HZ/100，得到Alpha使用的不同的HZ值。（即`timeout=2;`變成`timeout=2*HZ/100;`）

——把所有I/O內存（從640k到1MB）的指針引用替換為相應的readb() writeb() readl() writel()調用，如下例所示。

```
-       int *mem_base = (int *)dev->mem_start;
-       mem_base[0] = 0xba5eba5e;
+       unsigned long mem_base = dev->mem_start;
+       writel(0xba5eba5e, mem_base);
```

——把所有使用I/O內存作為源或目的地址的memcpy()調用替換為相應的`memcpy_fromio()`或者`memcpy_toio()`調用。

以結構無關的方式處理內存訪問的細節在近期的內核所附的文件`linux/Documentation/IO-mapping.txt`中進行了說明。

2.8.13 SUN/Sparc硬件上的Linux以太網卡。

要得到最新的Sparc信息，可以訪問以下URL：

Linux Sparc <http://www.geog.ubc.ca/sparc>

注意，有些Sparc的以太網硬件從主機獲得其MAC地址，因此可能會有多個接口具有相同的MAC地址。如果想在同一個網絡上使用多個接口，可以使用`ifconfig`的`hw`選項以分配唯一的MAC地址。

把PCI驅動程序移植到Sparc平台上與上面提到的AXP平台相似。可能的差異出在endian上，因為Sparc是big endian，而AXP和ix86是little endian。

2.8.14 其它硬件上的Linux以太網卡。

還有一些其它硬件平台可以運行Linux，比如Atari/Amiga（m68k）。就象Sparc一樣，最好是訪問每個Linux支持的平台主頁，以了解當前都支持哪些硬件。（歡迎提供這樣的站點連接——把它們發給我！）

2.8.15 不使用Hub連接10/100BaseT

可以不使用Hub連接基于10/100BaseT（RJ45）的系統嗎？

如果不使用額外的設備或機械裝置，可以很容易地連接兩台這樣的機器， 但不可能再多。參閱5.2 (雙絞線) ——解釋了如何做到這一點。而且你不可能簡單地交叉几根線或其它什么 就弄出一個Hub，不復制Hub也無法正確完成沖突信號。

2.8.16 SIOCSIFxxx: No such device

在啓動時出現了一大堆“SIOCSIFxxx: No such device”信息，后面還有一條“SIOCADDRT: Network is unreachable”，怎么回事？

你的以太網設備在啓動/插入模塊時沒有被檢測到，當`ifconfig`和`route`運行時，它們沒有可用的設備。使用`dmesg | more`來瀏覽啓動信息，看看有沒有檢測以太網卡的信息。

2.8.17 SIOCSFFLAGS: Try again

在運行“ifconfig”時出現“SIOCSFFLAGS: Try again”——怎么回事？

某些其它的設備使用了以太網卡想用的IRQ，所以以太網卡無法使用該IRQ。你不必重新啓動來解決這個問題，因為某些設備只是在需要時才獲取IRQ，在完成后就釋放了。例如某些聲卡、串口、軟盤驅動器等。你可以鍵入`cat /proc/interrupts`來看看哪些中斷正在 被使用。絕大多數Linux以太網卡驅動程序只有在用“ifconfig”打開時才獲取IRQ。如果你能讓其它設備“放開”所需的IRQ中斷線，那么你就可以用ifconfig來“Try again”了。

2.8.18 使用“ifconfig”得到的連接為UNSPEC，而硬件地址是00:00:00:00:00:00

在不帶參數運行ifconfig時，報告說連接為UNSPEC（而不是10Mbs以太網），而且硬件地址都是零。

是因為運行的“ifconfig”程序版本比內核的版本高。在與老版本的內核一起運行時，新版本的ifconfig無法報告這些特性。你可以升級內核，或者“降級”ifconfig，或者干脆不理會這個錯誤。內核知道硬件地址，所以即使ifconfig無法讀出它也沒有關系。

如果使用的`ifconfig`程序比使用的內核舊很多的話，也會出現一些奇怪的信息。

2.8.19 大量的RX和TX錯誤

在不帶參數運行ifconfig時，報告大量的接收和發送數據包錯誤。但看起來工作正常——怎么回事？

再看一遍。報告是說`RX packets` *big number* 停頓`errors 0` 停頓 `dropped 0` 停頓 `overrun 0`。所以你看到的那個大數字是機器接收和發送的數據包總數。如果還覺得不可思議，鍵入`cat /proc/net/dev`看看。

2.8.20 `/dev/`下的以太網卡入口

/dev/eth0像是個到/dev/xxx的連接。這樣對嗎？

與你聽過的正好相反，/dev/*下的文件沒被使用。你可以刪除掉任何 `/dev/wd0`、`/dev/ne0`以及類似的入口。

2.8.21 Linux與“trailers”

在“ifconfig”網卡時，需要禁止trailers嗎？

不能禁止trailers，而且也沒必要。“trailers”是避免在網絡層復制數據的工具。其想法是使用一個大小為“H”的固定大小的頭，把可變大小的頭信息放在包的尾部，并把所有包定位在頁開始之前的“H”字節。這只是個好想法，在實際中工作得并不好。如果有人建議使用“-trailers”，那不過是找個替罪羊罷了。這對解決問題沒有任何意義，但如果問題真的自行解決了，那么他就可以吹噓自己的神奇本領了。

2.8.22 訪問原始以太網設備

在Linux下怎樣不通過TCP/IP之類的東西訪問原始的以太網設備？

```
        int s=socket(AF_INET,SOCK_PACKET,htons(ETH_P_ALL));
```

這樣就可以得到一個接收所有協議類型的socket。對它執行`recvfrom()`調用，它就會用sa_family里的設備類型和sa_data數組里的設備名來填充sockaddr。我不知道是誰最早在Linux里使用SOCK_PACKET，但它確實是個非常好的東西。你也可以通過`sendto()`調用發送原始數據包。當然，在這樣做時你必須擁有root的權限。

3 性能小技巧

如果你的以太網吞吐量太小，或者想提高ftp傳輸的速度，可以使用這里的一些小技巧。

程序`ttcp.c`是測試原始吞吐量大小的好實驗。另一個普通的技巧是把一個大于1MB的`large_file`駐留在發送機器的緩沖內存里，然后執行`ftp> get large_file /dev/null`。（至少要“get”兩次，因為第一次是把文件駐留在發送機器的緩沖內存中。）把文件駐留在緩沖內存的原因是你對測量里混進從磁盤存取文件的速度不感興趣。這也是把輸入數據發送到`/dev/null`而不是磁盤上的原因。

3.1 一般概念

即使一個8比特的網卡在接收背靠背的數據包時也沒有問題。但在計算機無法迅速地從網卡接收數據包為更多進來的數據包騰出空間時，困難就出現了。如果計算機不能迅速地清除已經接收到的數據包占據的網卡內存，網卡就沒有空間存放新的數據包了。

在這種情況下，網卡或者丟棄新的數據包，或者把它放在以前接收的數據包位置上。任何一個都會引起重新傳送而嚴重地中斷平滑的數據傳輸流，并使性能被嚴重降低，其降級因子甚至可以達到5！

擁有更多板上內存的網卡可以“緩沖”更多的數據包，因此可以無需丟棄就處理更大量的突發背靠背數據包。這也就意味著網卡不需要主機響應從緩存讀取數據包的延遲更小以避免丟棄數據包。

大多數8比特網卡有8kB緩存，而大多數16比特網卡有16kB緩存。大多數的Linux驅動程序要保留3kB的緩存（作為兩個Tx緩存），這樣就只給8比特網卡剩下5kB作為接收的空間。它只夠存放三個完全大小（1500字節）的以太網數據包。

3.2 ISA網卡和ISA總線速度

在上面提到，如果從網卡移走數據包的速度足夠快，即使Rx數據包緩沖內存的數量小，丟棄/覆蓋的情況也不會出現。設置數據包從網卡移到計算機內存的速率的因子是連接這二者的數據通道的速度——即ISA總線的速度。（如果CPU是很慢的386sx-16，它也會起一定的作用。）

推荐的ISA總線時鐘為8MHz，但很多主板和周邊設備可以運行在更高的頻率上。一般通過在CMOS設置里選擇主板/CPU時鐘頻率因子，可以設置ISA總線的時鐘頻率。有些ISA和PCI/ISA主板可能沒有這一選項，那么就只能使用制造商的缺省值了。

例如，下面是在一台40MHz的486上，不同ISA總線速度下，8比特WD8003EP網卡用TTCP程序測量得到的一些接收速度。

```
ISA總線速度(MHz)    Rx TTCP (kB/s)
-------------------                  --------------
6.7                     740
13.4                    970
20.0                    1030
26.7                    1075
```

使用TCP/IP，對任何10Mb/s以太網卡都難以得到高于1075kB/s的結果。但是，不要期望每一個系統都可以工作在快的ISA總線速度下。大多數系統在高于13MHz的速度上無法正常工作。（同樣，某些PCI系統的ISA總線速度固定在8MHz，所以用戶沒有機會提高它。）

另外，對于快的傳輸速度，可以從較短的内存和I/O周期減少CPU占用量中獲益。（注意，設置在ISA總線上的硬盤和視頻卡也會從提高的ISA總線速度中感受到性能的提高。）

在實驗超過8MHz的ISA總線速度之前，一定要備份你的數據，并在提高速度后徹底測試所有的ISA周邊設備是否工作正常。

3.3 設置TCP的Rx窗口

再說一遍，網卡的板上RAM小以及網卡與計算機内存間的數據通道相對較慢會引起麻煩。TCP缺省的Rx窗口設置為32kB，也就是說與你在同一子網的一台較快的計算機會一下子發給你32kB的數據，而不會停下來看看你是否正常地接收到了它們。

近期版本的`route`命令能夠在運行過程中設置這個窗口的大小。一般它只用在需要減小該窗口的局域網，因為在若干路由器或網關后面的計算機會得到足夠的“緩沖”而不造成問題。下例為一個可能的用法：

```
        route add <whatever> ... window <win_size>
```

其中的`win_size`是你希望使用的窗口大小（以字節計算）。一個運行在8MHz或更低速度下的ISA總線上的8比特3c503網卡，可以在4kB大小的窗口下很好地工作。太大的窗口會導致覆蓋和丟棄數據包，并嚴重降低以太網的吞吐量。你可以用`cat /proc/net/dev`檢查操作狀態，它會顯示出現的丟棄或覆蓋狀況。

3.4 增强NFS性能

有些人發現在使用8kB（Sun的原始值）的NFS數據包大小時，在NFS客戶端使用8比特網卡得到的性能比預期的要差。

可能的原因在于8比特網卡與16比特網卡的板上緩存大小不同。最大的以太網數據包大小為1500字節。這樣8kB的NFS數據包就成為大約6個背靠背的最大大小的以太網數據包。8比特網卡和16比特網卡在接收背靠背數據包時都沒有問題。在機器無法及時從網卡緩存移走數據包時就有問題了，緩存會溢出。8比特網卡每次傳送時要多占用一個ISA總線周期的事實也幫不上忙。如果用的是8比特網卡，你能做的是把NFS傳送大小設置為2kB（甚至1kB），或者試著提高ISA總線速度以加快清空網卡上的緩存。我發現8MHz下（沒有其它系統負載），一個舊的WD8003E網卡可以跟得上2kB大小的NFS接收，但在4kB時，性能降低的因子為3。

另一方面，如果缺省的安裝選項為1kB大小，而你的網卡至少為16比特的 ISA網卡，把其大小提高到4kB（甚至8kB）會使性能顯著提高。

4 有關銷售商/制造廠家/型號的信息

下面按照銷售商名稱和產品標識的字母順序列出了許多網卡。在每個產品ID旁都標明了“支持”、“半支持”或“不支持”。

支持意味著該網卡的驅動程序存在，而且許多人在愉快地使用著此驅動程序，它看起來相當可靠。

半支持意味著該網卡的驅動程序存在，但至少下面的描述中有一個是真實的：(1) 驅動程序或硬件有Bug，會導致性能差、連接失敗甚至死機。(2) 驅動程序是新的或網卡不夠普及，因此驅動程序得到

的使用/測試機會很少，驅動程序的作者收到很少的反饋信息。很明顯(2)比(1)要好一些，在網卡/驅動程序的獨立描述中會指明它屬于哪種情況。對這兩種情況，在運行`make config`回答“Prompt for development and/or incomplete code/drivers?”時都應該回答“Y”。

不支持意味著該網卡目前還沒有可使用的驅動程序。這可能是由于對很少見的硬件缺乏興趣，也可能是銷售商還沒有發布編寫驅動程序所需的硬件資料。

注意，“支持”和“半支持”之間的區別是相當主觀的，是基于新聞組和郵件列表信息上得到的用戶反饋進行區分的。（無論如何，一個人來測試每個版本內核上所有網卡的所有驅動程序是不可能的！！！）所以你可能會發現一個被標為半支持的網卡工作得很好（這樣很不錯），或者一個被標為支持的網卡給你帶來無窮的麻煩（這樣就不夠好）。

在這些狀態后面，列出了Linux內核中給出的驅動程序名稱。這也是`/etc/conf.modules`模塊配置文件里`alias eth0 driver_name`一行所使用的驅動程序模塊的名稱。

4.1 3Com

如果你確定不了你的網卡是什么，但認為它是3Com網卡，你可以通過集成塊編號來識別它。3Com有一本資料“通過集成塊編號識別適配器”（ref 24500002），它應該能讓你弄明白。參見7.6 (3Com的技術信息)以了解如何從3Com索取該資料。

注意，3Com還有個有各種好東西的WWW/FTP站點：`ftp.3Com.com`，你也許想去看看。對于某些他們的網卡甚至還有Linux驅動程序，可能你想檢驗一下。據説他們的驅動程序不太穩定，或者在SMP和非基于ix86的機器上不穩定，所以需要記住這一點。

4.1.1 3c501

狀態：半支持，驅動程序名稱：3c501

這種過時的石器時代的8比特網卡用起來會很頭疼的。象瘟疫一樣躲開它吧。不要買這種網卡，哪怕是開玩笑也不要買。它的性能很差勁，而且很容易壞掉。

對那些還不肯相信的人來説，3c501一次只能干一件事——在從其單包緩存移出一個數據包時，它不能接收另一個包，在載入一個發送包時也不能接收一個包。對于兩台處理一個包并回答會花掉10毫秒的基于8088的計算機之間的網絡而言，這就不錯了，但現代網絡几乎每次事務都會發送背靠背數據包。

AutoIRQ可以工作，沒使用DMA，自動檢測只能查看`0x280`和`0x300`，診斷級別通過第三個啓動參數設置。

再説一遍，强烈反對使用3c501！更差的是在IP廣播内核中，在偵聽所有廣播包時你會進入暫停狀態。閱讀源碼頭部的注釋以得到更多的細節。

4.1.2 EtherLink II, 3c503, 3c503/16

狀態：支持，驅動程序名稱：3c503 (+8390)

3c503沒有“EEPROM setup”，所以在Linux下運行該網卡無需使用診斷/設置程序。3c503的共享内存地址是用跳線設置的，與啓動PROM地址共享。熟悉其它ISA網卡的人可能會被弄糊涂，因為其它ISA網卡除非有啓動PROM，都是讓跳線設為“禁止”的。

因為WD80x3的總線寬度一樣，這些網卡的速度也應該一樣，但實際上更慢一些。這些共享内存的的以太網卡還有一種不使用8390工具的可編程I/O模式（它們的工程師發現了太多的Bug！）。Linux的

3c503驅動程序也可以工作在3c503的可編程I/O模式，但不如共享內存模式可靠，而且更慢。同樣，在更新驅動程序時可編程I/O模式沒有很好地測試。不要使用可編程I/O模式，除非要與MS-DOS兼容。

3c503的IRQ中斷線用軟件設置，在EEPROM中沒有其痕跡。與MS-DOS驅動程序不同，Linux的驅動程序有autoIRQ的能力：它使用{5,2/9,3,4}中第一條可用IRQ中斷線，在每次運行ifconfig時進行選擇。（老式的驅動程序版本在啓動時選擇IRQ。）如果沒有IRQ中斷線可用，“ifconfig”里的ioctl()調用將返回EAGAIN。

某些使用503會遇到的常見問題在2.7 (3Com網卡的問題)中進行了討論。

如果你想把該驅動程序作為一個可載入的模塊，應該閱讀9.2 (把以太網驅動程序作為模塊使用)以了解有關模塊的信息。

注意，一些舊的無盤386工作站有一個板上的3c503（3Com制造，但用其它名稱銷售，如“Bull”），但銷售商ID不是3Com的ID，所以檢測不出來。更多的資料可以在Etherboot軟件包里找到，要啓動這些無盤工作站你一定得有這些東西。

4.1.3 Etherlink Plus 3c505

狀態：半支持，驅動程序名稱：3c505

這些網卡使用i82586芯片，但沒有太多的有關這些網卡的資料。標准內核包含該驅動程序，但它被歸為alpha版本的驅動程序。參見2.2 (Alpha驅動程序)以了解在Linux下使用alpha/測試版本以太網驅動程序的重要信息。

如果你要使用這樣的網卡，還應該讀一下`/usr/src/linux/drivers/net/README.3c505`文件。它包含了你可以使用/禁止的各種選項。

4.1.4 Etherlink-16 3c507

狀態：半支持，驅動程序名稱：3c507

該網卡使用了一塊Intel芯片，其驅動程序的開發與Intel的Ether Express驅動程序開發關系很密切。在標准的內核發行版本中包含了其驅動程序，但是是作為一個alpha版本的驅動程序。參見2.2 (Alpha驅動程序)以了解在Linux下使用alpha/測試版本以太網驅動程序的重要信息。

4.1.5 Etherlink III, 3c509 / 3c509B

狀態：支持，驅動程序名稱：3c509

這種網卡相當便宜，而且對ISA的無總線控制的設計而言性能相當好。其缺點在于原始的3c509要求很低的中斷延遲。3c509B有一個較大的緩存，應該沒有這個問題。（見下文。）這些網卡使用PIO傳輸，與ne2000網卡相似，相對而言，wd8013一類使用共享內存的網卡效率更高一些。

原始的3c509有一個小的包緩存（總共4kB，2kB Rx，2kB Tx），如果中斷屏蔽的時間過長，驅動程序就會丟棄數據包。要減輕這個問題，可以嘗試在IDE磁盤傳輸過程中不屏蔽中斷（參見`man hdparm`），或者提高ISA總線速度使IDE傳輸結束得早一些。

新型號的3c509B板上緩存為8kB，并且可以把Rx/Tx划分成4/4、5/3或6/2。這一設置要用DOS配置工具來改變，并保存在EEPROM上。這會減輕上面提到原始3c509的問題。

3c509B的用戶可以使用提供的DOS工具來禁止即插即用支持，并把輸出媒介設置為要求的東西。目前的Linux驅動程序還不支持自動檢測媒介設置，所以必須選擇10Base-T、10Base-2或AUI。注意，要完

全禁止PnP，需要使用`3C5X9CFG /PNP:DISABLE`并重新冷啓動以確保設置生效。

些人詢問DOS配置工具中出現的“Server or Workstation”和“Highest Modem Speed”設置。Donald寫道：“這些只不過是給驅動程序的提示，而Linux驅動程序不使用這些參數：它總是優化為高吞吐量，而非低延遲（‘Server’）。低延遲對于老式的無窗口IPX吞吐量非常關鍵。為了減少延遲，3c509的MS-DOS驅動程序對某些操作禁止中斷，阻塞串口中斷。因此需要‘modem speed’設置。Linux驅動程序只對整個包進行操作，例如在數據包完全傳到網卡后才開始發送該數據包，從而避免了長時間禁止中斷的需要。”

注意，ISA網卡檢測使用的方法與大多數網卡不同。基本上是通過發送數據到ID_PORT（從端口`0x100`到`0x1ff`，間隔為`0x10`）要求網卡響應。這一檢測方法意味著在多ISA的3c509配置中，某個特定網卡總是首先被檢測，而硬件以太網地址最低的網卡總是作為`eth0`。這其實沒什么問題，但有些人想為特定的接口分配一個6字節的硬件地址。如果有多塊3c509網卡，最好不指定I/O端口（即使用I/O=zero）使用`ether=0,0,ethN`命令，讓自動檢測來排列哪一塊網卡作為第一塊網卡。使用非零的I/O值會使該命令無法檢測所有的網卡，所以不要這樣做。

如果這使你感到困擾，看一下Donald的最新驅動程序，你可以在未使用的內存地址域中使用一個`0x3c509`值，使檢測的順序符合你的要求。

4.1.6 3c515

狀態：支持，驅動程序名稱：3c515

這是3Com的ISA 100Mbps產品，編碼名為“CorkScrew”。v2.2內核里包含了從Donald那兒得到的這些網卡的相對較新的驅動程序。要獲得更新的資料，可以訪問Vortex的網頁：

Vortex `<http://www.scyld.com/linux/drivers/vortex.html>`

4.1.7 3c523

狀態：半支持，驅動程序名稱：3c523

這個MCA總線的網卡使用i82586，Chris Beauregard修改了ni52驅動程序用于這些網卡。在v2.2內核源碼樹里可以找到它的驅動程序。

更多的細節可以在MCA-Linux的網頁上找到 `http://glycerine.cetmm.uni.edu/mca/`

4.1.8 3c527

狀態：不支持。

對，這是另一個MCA網卡，對它沒興趣。如果你堅持用MCA，最好用3c529。

4.1.9 3c529

狀態：支持，驅動程序名稱：3c509

這個網卡實際上使用與3c509相同的芯片組。實際上早在內核里加進對MCA的支持之前，Donald就已經在3c509的驅動程序里放了些鉤子函數，在探測完EISA網卡后，開始探測ISA網卡之前，檢查MCA網卡。所需的MCA探測代碼被包含在與v2.2內核一起提供的驅動程序里。更多的細節可以在MCA-Linux網頁上找到：

`http://glycerine.cetmm.uni.edu/mca/`

4.1.10 3c562

狀態：支持，驅動程序名稱：3c589（獨立發行）

這個PCMCIA網卡是3c589B網卡與modem的組合。對于終端用戶，modem與一個標准的modem一樣。唯一的麻煩是要使兩個獨立的Linux驅動程序共用一個中斷。該卡上有一些新的寄存器和一些支持共享中斷的硬件。需要使用支持共享中斷的v2.0以上的內核版本。

再次感謝Cameron提供給David Hinds的樣品和資料。請在David發行的PCMCIA軟件包里尋找支持。

參見8.2 (PCMCIA支持)以了解更多的有關PCMCIA芯片組、socket啓用軟件等信息。

4.1.11 3c575

狀態：未知。

這個PCMCIA網卡的驅動程序正在開發中，但愿以后會包含在David的PCMCIA軟件包里。最好檢查一下PCMCIA軟件包以了解目前的狀態。

4.1.12 3c579

狀態：支持，驅動程序名稱：3c509

EISA版本的509。目前的EISA版本還使用同樣的16比特寬的芯片，而不是32比特的接口，所以性能上的改善并不令人驚奇。確定網卡被配置為EISA尋址模式。閱讀上面的3c509一節以了解驅動程序的信息。

4.1.13 3c589 / 3c589B

狀態：半支持，驅動程序名稱：3c589

現在這種PCMCIA網卡已經被很多人使用了相當長時間。注意對它的支持（目前）還不包括在缺省的內核源碼樹內。名稱中“B”的含義與3c509的情況相同。

在Donald的ftp站點和David Hinds的PCMCIA軟件包里都有可用的驅動程序。你還需要一套支持PCMCIA控制器的芯片組。參見8.2 (PCMCIA支持)以了解更多的有關PCMCIA芯片組、socket啓用軟件等信息。

4.1.14 3c590 / 3c595

狀態：支持，驅動程序名稱：3c59x

這些“Vortex”網卡是用在PCI總線機器上的，’590是10Mbps的產品，’595是3Com的100Mbs產品。注意’595也可以象’590一樣運行（即使用10Mbps模式）。驅動程序包含在v2.0內核源碼中，但還在不斷升級。如果在使用v2.0內核里的驅動程序時遇到問題，可以從下面的URL獲取一個更新的驅動程序：

Vortex `<http://www.scyld.com/linux/drivers/vortex.html>`

注意，現在有兩種不同的3c590網卡，較早的型號有32kB的板上內存，后來的型號只有8kB的板上內存。可能你買一個新的3c59x還沒有多久，它就正在被3c90x所取代。如果你是買別人的二手貨，想法弄到一個32kB的型號。3c595網卡有64kB板上內存，因為無法只用8kB的RAM就能達到100Mbps！

感謝Cameron Spitzer和3Com的Terry Murphy給Donald提供網卡和資料，使Donald能寫出驅動程序來。

4.1.15 3c592 / 3c597

狀態：支持，驅動程序名稱：3c59x

些是EISA版本的3c59x系列網卡。3c592/3c597（又名為Demon）可以使用上面討論的vortex驅動程序。

4.1.16 3c900 / 3c905 / 3c905B / 3c905C

狀態：支持，驅動程序名稱：3c59x

這些網卡（又名為“Boomerang”，或EtherLink III XL）是用來取代3c590/3c595網卡的。

對Cyclone的“B”版本的支持最近才加進來。要在v2.0之前的內核使用該網卡，必須從如下的Donald站點獲取更新后的3c59x.c驅動程序：

Vortex-Page <http://www.scyld.com/network>

4.1.17 3c985

狀態：支持，驅動程序名稱：acenic

Jes Sorensen提供的驅動程序在v2.2內核里能找到。除了3Com的型號，它還支持几種其它的吉比特網卡。

4.2 Accton

4.2.1 Accton MPX

狀態：支持，驅動程序名稱：ne (+8390)

不要被它的名字騙了。它還是NE2000兼容網卡，可以使用ne2000驅動程序。

4.2.2 Accton EN1203, EN1207, EtherDuo-PCI

狀態：支持，驅動程序名稱：de4x5, tulip或rtl8139

顯然有几種修訂版的EN1207（A到D），其中A、B和C是基于tulip的，而D版本的是基于RealTek 8139（驅動程序不同）。所以在購買時要確定如果出問題可以退回。

4.2.3 Accton EN2209 Parallel Port Adaptor (EtherPocket)

狀態：半支持，驅動程序名稱：?

現在有這些并口適配器的驅動程序，但還不包括在2.0或2.1版本內核源碼中。可以從如下主頁獲取驅動程序：

http://www.unix-ag.uni-siegen.de/~nils/accton_linux.html

4.2.4 Accton EN2212 PCMCIA Card

狀態：半支持，驅動程序名稱：?

David Hinds在編寫該網卡的驅動程序，最好檢查一下他的PCMCIA軟件包以了解目前的狀態。

4.3 Allied Telesyn/Telesis

4.3.1 AT1500

狀態：支持，驅動程序名稱：lance

這些是使用79C960版本的AMD LANCE的系列低成本以太網卡。它們是可以控制總線的網卡，因此是一種可用的較快的ISA總線以太網卡。

DMA選擇和芯片號碼信息可以在4.4.1 (AMD LANCE)找到。

更多有關基于AMD LANCE的以太網卡的技術信息可以在7.7 (Notes on AMD...)找到。

4.3.2 AT1700

狀態：支持，驅動程序名稱：at1700

注意，要在`make config`時訪問該驅動程序依然需要先在回答"Prompt for development and/or incomplete code/drivers?"時回答"Y"。這只是因為這種網卡很少見，導致驅動程序穩定性的反饋太少。如果與內核一起提供的驅動程序出現問題，也許你會對以下的替代驅動程序感興趣：`http://www.cc.hit-u.ac.jp/nagoya/at1700/`

Allied Telesis的AT1700系列網卡是基于Fujitsu的MB86965芯片的。該芯片使用可編程I/O接口和一對固定大小的傳輸緩存。這樣在發送背靠背的一小組數據包時，切換緩存帶來的停頓較小。

該網卡的一個獨特之處在于除了能夠驅動10baseT的100歐姆UTP（非屏蔽雙絞線）之外，還能夠驅動一般安裝在令牌環上的150歐姆STP（屏蔽雙絞線）。該網卡還有一個光纖版本(AT1700FT)。

用在AT1700上的Fujitsu芯片有一個設計上的缺陷：它需要機器斷電來完全重起。使用reset按鈕無法重起總線接口。這也不是太壞，只不過在重起后才能可靠地被檢測到。解決方法是在內核檢測AT1700有問題時關機器重起。

4.3.3 AT2400

狀態：支持，驅動程序名稱： ne, ne2k-pci (+8390)

還是PCI NE2000兼容網卡，它是基于RealTek 8029芯片的。

4.3.4 AT2450

狀態：支持，驅動程序名稱：pcnet32

這是PCI版本的AT1500，而且沒有Boca的79c970 PCI網卡那樣的問題。DMA選擇和芯片號碼信息可以在4.4.1 (AMD LANCE)找到。

更多有關基于AMD LANCE的以太網卡的技術信息可以在7.7 (Notes on AMD...)找到。

4.3.5 AT2500

狀態：半支持，驅動程序名稱：rtl8139

網卡使用RealTek的8139芯片——參見4.31.5 (RealTek 8139)一節。

4.3.6 AT2540FX

狀態：半支持，驅動程序名稱：eepro100

該網卡使用i82557芯片，因此可能/應該能夠使用eepro100的驅動程序。如果你試用過，請發給我一份報告以更新本信息。

4.4 AMD / Advanced Micro Devices

AMD的Carl Ching友好地提供了非常詳盡的所有有關AMD以太網產品的資料，對澄清本節內容很有幫助。

4.4.1 AMD LANCE (7990, 79C960/961/961A, PCnet-ISA)

狀態：支持，驅動程序名稱：lance

這實際上不是AMD的以太網卡。你會閱讀本小節的原因是，你在網卡上只能找到AMD標記和上述數字。7990是原始的“LANCE”芯片，但大多數資料（包括本文檔）都按照“LANCE”芯片對待所有這些類似的芯片。（......我可以加一句：這樣是不對的。）

上面這些數字指明了作為許多以太網卡的核心的AMD芯片。例如，Allied Telesis的AT1500（參見4.3.1 (AT1500)）和NE1500/2100（參見4.27.4 (NE1500)）就使用了這些芯片。

7990/79c90早已被新版本所取代。79C960（又名PCnet-ISA）主要包含79c90核心以及所有其它必須的硬件支持，可作為單片以太網卡的解決方案。79c961(PCnet-ISA+)是'960的無跳線即插即用版本。ISA系列的最后一個芯片是79c961A(PCnet-ISA II)，增加了全雙工的功能。除了那些使用共享內存配置的原始7990的非常古老的網卡，所有使用這些芯片的網卡都可以使用lance.c驅動程序。可以通過DMA通道沒有跳線來辨認出這些老網卡。

一個常見的問題是遇到“busmaster arbitration failure”消息。在固定的時間之內（50us）LANCE驅動程序無法訪問總線時就會出現這一消息。這通常預示主板上的總線控制DMA被破壞，或者某些其它設備霸占了總線，或者DMA通道出現沖突。如果你的BIOS設置中有“GAT option”（保証存取時間），那么試著改變一下其設置看看有沒有效。

同樣要注意，對一個合法網卡，驅動程序只查看以下地址：`0x300`，`0x320`，`0x340`，`0x360`，啓動參數`ether=`所提供的地址被靜悄悄地略過了（對此將進行修改），所以現在要確定你的網卡配置為上述的一個I/O地址。

即使安裝了超過16MB的內存，該驅動程序依然工作得很好，因為必要時它會使用低端內存“bounce-buffers”（即在發送數據到網卡進行傳輸之前，16MB以上內存的數據會先拷貝到16MB以下的一個緩存里。）

DMA通道可以用除此之外不被使用的dev->mem_start值的低位進行設置（又名PARAM_1）。（參見9.1.1 (PARAM_1)）如果沒有設置，那么按次序啓用每個空閑的DMA通道并檢查是否初始化成功來進行探測。

HP-J2405A板是個例外：在該板上很容易讀出EEPROM設置的IRQ和DMA的值。

見7.7 (Notes on AMD...)以了解更多有關這些芯片的信息。

4.4.2 AMD 79C965 (PCnet-32)

狀態：支持，驅動程序名稱：pcnet32

這是PCnet-32——用于VL總線和局部總線系統的原始LANCE芯片的32比特總線控制版本。這些芯片可以用標准的`lance.c`驅動程序操作，同時還有一個不受任何ISA總線有關16MB限制的32比特版本的驅動程序（`pcnet32.c`）。

4.4.3 AMD 79C970/970A (PCnet-PCI)

狀態：支持，驅動程序名稱：pcnet32

這是PCnet-PCI——與PCnet-32相似，不過是為基于PCI總線系統設計的。請參閱上面的PCnet-32信息。這意味著需要構建一個啓用PCI BIOS支持的內核。'970A在原始的'970設計上增加了全雙工支持及其它一些特性。

注意，Boca制造的79C970無法用在較快的奔騰機器上。這是個硬件錯誤，它也會影響DOS用戶。參考Boca一節以了解更多的細節。

4.4.4 AMD 79C971 (PCnet-FAST)

狀態：支持，驅動程序名稱：pcnet32

這是用于PCI系統的AMD的100Mbit芯片，也支持全雙工操作。1996年6月出產。

4.4.5 AMD 79C972 (PCnet-FAST+)

狀態：支持，驅動程序名稱：pcnet32

已經証實該網卡工作情況與'971一樣。

4.4.6 AMD 79C974 (PCnet-SCSI)

狀態：支持，驅動程序名稱：pcnet32

這是PCnet-SCSI——從以太網的觀點來看基本上按'970處理。參看上文。不要問是否支持芯片的SCSI部分——本文是以太網-*HowTo*，不是SCSI-HowTo。

4.5 Ansel Communications

4.5.1 AC3200 EISA

狀態：半支持，驅動程序名稱：ac3200

注意，要在`make config`時訪問該驅動程序依然需要先在回答"Prompt for development and/or incomplete code/drivers?"時回答"Y"。這只是因為這種網卡很少見，導致驅動程序穩定性的反饋太少。

其驅動程序作為alpha測試驅動程序被包含在目前的内核里。該網卡是基于ne2000和wd80x3網卡上使用的普通的NS8390芯片。請參見本文的2.2 (Alpha驅動程序)以了解在Linux下使用alpha版驅動程序的重要信息。

如果你在使用該網卡，請讓我們知道它的工作情況，雖然在v1.1.25内核里就包含了其驅動程序，但反饋信息很少。

如果你想把該驅動程序作為一個可載入的模塊，應該閱讀9.2 (把以太網驅動程序作為模塊使用)以了解有關模塊的信息。

4.6 Apricot

4.6.1 Apricot Xen-II On Board Ethernet

狀態：半支持，驅動程序名稱：apricot

這是使用i82596總線控制芯片的板上以太網卡。它的I/O地址只能是`0x300`。從驅動程序的源碼上得知，其IRQ也被硬件固定為10。

早期版本的驅動程序傾向于認為`0x300`處存放的一定是apricot NIC。后來才對硬件地址進行檢查以避免錯誤的檢測。

4.7 Arcnet

狀態：支持，驅動程序名稱：arcnet (arc-rimi, com90xx, com20020)

由于該網卡的價格非常低廉而且以太網性能很好，可能人們會有很多機會捐獻出他們的Arcnet硬件，使得很多家用系統都使用Arcnet。

Arcnet的一個優點在于所有的網卡接口都一樣，所以只需要一個驅動程序。它還内建了錯誤處理機制，據信不會丢失數據包。（對UDP來說這可太棒了！）注意，arcnet驅動程序使用“arc0”作為自己的名字，而不是通常的以太網設備名“eth0”。

在標准内核里包含了若干設置跳線、一般提示和郵寄Bug報告地址的信息文件。

據信100Mbs的ARCnet網卡也可以同樣使用該驅動程序！

4.8 AT&T

注意，AT&T的StarLAN是一種孤島技術，跟SynOptics的LattisNet一樣，不能在標准的10Base-T環境里使用，沒有能進行“相互交流”的Hub。

4.8.1 AT&T T7231 (LanPACER+)

狀態：不支持。

這些StarLAN網卡使用的接口類似于i82586芯片。Matthijs Melchior（`matthijs.n.melchior@att.com`）曾經試著用過3c507的驅動程序，好象差不多可以工作了。但從那之后就再沒聽過多少消息了。

4.9 Boca Research

不錯，他們不光是制造多口串行卡。

4.9.1 Boca BEN400

狀態：支持，驅動程序名稱：ne (+8390)

顯然這是一款NE2000兼容網卡，使用VIA VT86C916芯片。

4.9.2 Boca BEN (ISA, VLB, PCI)

狀態：支持，驅動程序名稱：lance, pcnet32

這些網卡都是基于AMD的PCnet芯片。需要警告一下潛在的買家，許多用戶在使用這些VLB/PCI網卡時遇到了沒完沒了的麻煩。快的奔騰系統用戶受到了特別的打擊。注意這并不是驅動程序的問題，因為DOS/Win/NT用戶也一樣遇到麻煩。Boca的技術支持電話號碼是(407) 241-8088，也可以通過`75300.2672@compuserve.com`找到他們。老式的ISA網卡看來沒遇到這種問題。

Boca對受影響的使用者提供"包修"服務，需要增加一個缺少的電容，雖然可以解決一些問題，但對大多數人來說并非百分之百奏效。

如果您仍然想買這種網卡，至少試著得到一個7天無條件退貨的條款，這樣一旦出問題你就可以退貨。

有關AMD芯片更一般的信息可以在4.4.1 (AMD LANCE)找到。

基于AMD LANCE的以太網卡的更多技術信息可以在7.7 (Notes on AMD...)找到。

4.10 Cabletron

Donald寫道："是的，又一個不肯提供編程資料的公司。他們要等待几個月來最終確定 所有他們的資料都是私有的，蓄意浪費我的時間。如果可以的話，象躲避瘟疫一樣避免使用他們的網卡。還要注意，有人曾打電話給Cabletron，結果聽到什么'有個D. Becker在編寫Linux下的驅動程序'之類的話——好象我是為他們工作的。完全不是這么回事。"

看來自從几年前Donald寫了上述注釋后，Cabletron改變他們對于編程資料的政策（象Xircom一樣）——如果你想証實這一點或索取編程信息的話，發個e-mail給`support@ctron.com`。但是，現在几乎沒有人對老式的E20xx和E21xx網卡提出修改/更新驅動程序的要求了。

4.10.1 E10**, E10**-x, E20**, E20**-x

狀態：半支持，驅動程序名稱：ne (+8390)

這些差不多算是NEx000兼容網卡，據報告可以使用標准的NEx000驅動程序，這要感謝探測時的ctron特性檢查。如果有問題，几乎不可能修復，因為沒有編程資料。

4.10.2 E2100

狀態：半支持，驅動程序名稱：e2100 (+8390)

又來了，在編程資料私有的情況下什么都做不了。E2100的設計很差勁。每次傳送數據包映射它的共享內存時，它都要映射到整個*128K*的區域！這就意味著你無法在此區域安全地使用另一個中斷驅動的共享內存設備，包括另一個E2100。絕大多數時間它都正常工作，但時不時地會咬你一口。（對，這個問題可以通過傳送數據包時關中斷來避免，但那樣几乎肯定會丟失時鐘ticks。）同樣，如果錯誤地對板進行了編程，或者在錯誤的時刻暫停機器，即使reset按鈕都無法生效。你必須關機并保持30秒之后再開機。

媒介選擇是自動進行的，但可以用dev->mem_end參數的低比特位來覆蓋。參見9.1.1 (PARAM_2)。模塊用戶可以在`/etc/conf.modules`文件里指定`xcvr=N`值作為`option`。

同樣，不要把E2100當成NE2100的兼容網卡。E2100是采用NatSemi的DP8390的共享內存設計，大致與傷腦筋的WD8013相似，而NE2100（和NE1500）是用總線控制的AMD的LANCE設計的。

在標准內核里包含了E2100的驅動程序。但是，無法看到其編程資料，不要指望有Bug修復。除非你已經買了該網卡，不要用它。

如果你想把該驅動程序作為一個可載入的模塊，應該閱讀9.2 (把以太網驅動程序作為模塊使用)以了解有關模塊的信息。

4.10.3 E22**

狀態：半支持，驅動程序名稱：lance

按照Cabletron技術公告牌上的資料，這些網卡使用標准AMD的PC-Net芯片組（參見4.4.1 (AMD PC-Net)）而且可以使用普通的lance驅動程序。

4.11 Cogent

下面是進行聯系的地址和方法：

```
Cogent Data Technologies, Inc.
175 West Street, P.O. Box 926
Friday Harbour, WA 98250, USA.

Cogent Sales
15375 S.E. 30th Place, Suite 310
Bellevue, WA 98007, USA.

技術支持：
Phone (360) 378-2929 between 8am and 5pm PST
Fax (360) 378-2882
Compuserve GO COGENT
Bulletin Board Service (360) 378-5405
Internet: support@cogentdata.com
```

4.11.1 EM100-ISA/EISA

狀態：半支持，驅動程序名稱：smc9194

這些網卡使用SMC的91c100芯片，可以使用SMC的91c92驅動程序，但還沒得到証實。

4.11.2 Cogent eMASTER+, EM100-PCI, EM400, EM960, EM964

狀態：支持，驅動程序名稱：de4x5, tulip

些是另一種DEC的21040網卡，希望可以使用標准21040驅動程序很好地工作。

EM400和EM964是使用DEC的21050橋接器和4片21040芯片的4端口網卡。

參見4.16.4 (DEC 21040)以了解這些網卡的更多信息和當前驅動程序的狀況。

4.12 Compaq

Compaq并不真正從事制造以太網卡，但他們的很多系統都在主板上嵌入了以太網控制器。

4.12.1 Compaq Deskpro / Compaq XL (Embedded AMD Chip)

狀態：支持，驅動程序名稱：pcnet32

諸如XL系列的機器在主板上有一片AMD的79c97x PCI芯片，可以使用標准的LANCE驅動程序。但在使用前，必須使用一些技巧把PCI BIOS放到可以被Linux發現的地方。Frank Maas友好地提供了其中的細節：

"Compaq機器的這個問題在于其PCI目錄被載入高端內存，Linux內核無法達到那一點。結果：網卡檢測不到，也無法使用（另外鼠標也不工作）。其解決方法（在http://www-c724.uibk.ac.at/XL/里有詳盡的描述）是載入MS-DOS，使用Compaq編寫的一個小驅動程序，然后用LOADLIN載入Linux內核。好吧，你也許想說"呸、呸"，但到目前為止，這是所我知道的唯一辦法。那個小驅動程序的作用只不過是把PCI目錄放到其正常的存儲位置（這樣Linux就可以找到它了）"

更多有關AMD芯片的一般資料可以在4.4.1 (AMD LANCE)找到。

4.12.2 Compaq Nettelligent/NetFlex（嵌入ThunderLAN芯片）

狀態：支持，驅動程序名稱：tlan

這些系統使用Texas Instruments的ThunderLAN芯片。有關ThunderLAN驅動程序的資料可在4.36.1 (ThunderLAN)找到。

4.12.3 Compaq PCI card

狀態：支持，驅動程序名稱：eepro100

檢查你的網卡——如果它的零件登記號是323551-821或者上面有一片intel 82558芯片，那它就是另一種基于Intel EEPro100的網卡。

4.13 Danpex

4.13.1 Danpex EN9400

狀態：支持，驅動程序名稱：de4x5, tulip

又一個基于DEC的21040芯片的網卡，據報告性能挺好，價格比較便宜。

參見4.16.4 (DEC 21040)以了解這些網卡的更多信息和當前驅動程序的狀況。

4.14 D-Link

4.14.1 DE-100, DE-200, DE-220-T, DE-250

狀態：支持，驅動程序名稱：ne (+8390)

一些早期的D-Link網卡沒有PROM的標識`0x57`，但ne2000驅動程序能識別它們。對于軟件配置的網卡，可以從`www.dlink.com`得到配置程序。DE2**網卡曾經廣為報道在早期版本的Linux下會產生傳輸地址不匹配的虛假錯誤。注意，有一些Digital (DEC)的網卡也叫做DE100和DE200，只是名稱相同而已。

4.14.2 DE-520

狀態：支持，驅動程序名稱：pcnet32

這是使用PCI版本AMD的LANCE芯片的PCI網卡。DMA選擇和芯片號碼信息可以在4.4.1 (AMD LANCE)找到。

更多有關基于AMD LANCE的以太網卡的技術信息可以在7.7 (Notes on AMD...)找到。

4.14.3 DE-528

狀態：支持，驅動程序名稱：ne, ne2k-pci (+8390)

很明顯D-Link已經開始制造PCI NE2000兼容網卡了。

4.14.4 DE-530

狀態：支持，驅動程序名稱：de4x5, tulip

這是一款使用DEC的21040 PCI芯片的通用網卡，據報告可以使用通用的21040 tulip驅動程序。注意，它不是DFE-530。

參見4.16.4 (DEC 21040)以了解這些網卡的更多信息和當前驅動程序的狀況。

4.14.5 DE-600

狀態：支持，驅動程序名稱：de600

筆記本電腦以及想讓電腦快捷地聯上以太網的用戶可能用該網卡比較好。在缺省的內核源碼樹里有它的驅動程序，是Bjorn Ekwall `bj0rn@blox.se`編寫的。通過并口可以從該網卡獲得180kb/s的傳送速度。請閱讀內核源碼樹里的README.DLINK文件。

注意，現在傳遞給`ifconfig`的設備名為`eth0`，而不是以前用的`dl0`。

如果你的并口不是在標准的`0x378`，那么需要重新進行編譯，因為地址是直接編譯到驅動程序里的。同樣還要注意，某些筆記本電腦主板上的并口位于`0x3bc`，某些一體化板卡的并口也是這樣。

4.14.6 DE-620

狀態：支持，驅動程序名稱：de620

DE-600一樣，只是有兩種輸出格式。Bjorn為1.1以上版本內核里的這種型號網卡編寫了驅動程序。參見上面有關DE-600的信息。

4.14.7 DE-650

狀態：半支持，驅動程序名稱：de650 (?)

這種PCMCIA網卡已經被有些人在他們的筆記本電腦上使用一段時間了。它是個基本的8390設計，與NE2000很相象。LinkSys的PCMCIA網卡和IC-Card的Ethernet都被認為是DE-650兼容網卡。注意，目前其驅動程序還不是標准內核的一部分，所以需要做一些打補丁的工作。參見本文檔的8.2 (PCMCIA支持)。

4.14.8 DFE-530TX

狀態：支持，驅動程序名稱：via-rhine

又一款使用VIA Rhine芯片組。（參見4.38.2 (VIA Rhine)）不要把它同基于tulip的網卡DE-530相混淆。

4.14.9 DFE-538TX

狀態：支持，驅動程序名稱：rtl8139, 8139too

該網卡使用RealTek 8139芯片——參見4.31.5 (RealTek 8139)一節。

4.15 DFI

4.15.1 DFINET-300和DFINET-400

狀態：支持，驅動程序名稱：ne (+8390)

又一款拙劣設計的NE兼容網卡——這些網卡在PROM的前三個字節使用“DFI”，而不是象所有NE1000和NE2000網卡那樣在第14和15字節使用`0x57`。（300是8比特的偽NE1000兼容卡，400是偽NE2000兼容卡。）

4.16 Digital / DEC

4.16.1 DEPCA, DE100/1, DE200/1/2, DE210, DE422

狀態：支持，驅動程序名稱：depca

源文件“depca.c”里的説明包含了在一台機器里如何使用多塊此類網卡。注意，DE422是EISA網卡。這些網卡都是基于AMD的LANCE芯片。參見4.4.1 (AMD LANCE)以了解更多信息。同時最多只能使用兩塊ISA網卡，因為它們的I/O基址只能設置在`0x300`和`0x200`。如果想這么做的話，請閱讀標准內核源碼樹中的驅動程序源文件`depca.c`里的注釋。

其驅動程序也可以用在基于Alpha CPU的機器上，用戶還可以使用其中的各種ioctl()函數。

4.16.2 Digital EtherWorks 3 (DE203, DE204, DE205)

狀態：支持，驅動程序名稱：ewrk3

些網卡使用DEC的私有芯片，而不是象早期的DE200網卡那樣使用LANCE芯片。它們支持共享內存和可編程I/O兩種工作方式，但在使用PIO模式時其性能會下降50%。共享內存的大小可以設置為2kB、32kB或64kB，但驅動程序只進行了2kB和32kB下的測試。在驅動程序文件`ewrk3.c`的開頭有更多的信息（包括如何把它作為可載入模塊使用），在`README.ewrk3`里也有此類信息。在發行的標准內核里包含了這兩個文件。與depca.c一樣，該驅動程序也支持Alpha CPU。

標准的驅動程序里有一些有趣的ioctl()調用，可以用來獲取或清除數據包統計信息、讀/寫EEPROM、改變硬件地址等。黑客可以閱讀源代碼以了解更多此類信息。

除了其它工具軟件，David還為這些網卡編寫了一個配置工具（依照DOS程序`NICSETUP.EXE`）。這些都可以在大多數Linux的FTP站點的目錄`/pub/Linux/system/Network/management`下找到——查找文件`ewrk3tools-X.XX.tar.gz`。

4.16.3 DE425 EISA, DE434, DE435, DE500

狀態：支持，驅動程序名稱：de4x5, tulip

這些網卡是基于下面會提到的21040芯片。DE500使用21140芯片提供10/100Mbs以太網的連接。閱讀下面的21040一節以了解其它信息。在非DEC網卡使用此驅動程序時還有一些編譯時的選項。細節可見`README.de4x5`。

所有Digital網卡都自動探測媒介（除了DE500暫時因為專利的原因還不支持這一點）。

驅動程序也支持Alpha CPU，可以被作為模塊載入。用戶可以通過ioctl()調用訪問驅動程序的內部——參考“ewrk3”工具和de4x5.c源碼以了解如何使用。

4.16.4 DEC 21040, 21041, 2114x, Tulip

狀態：支持，驅動程序名稱：de4x5, tulip

DEC 21040是Digital的總線控制單片以太網卡產品，與AMD的PCnet芯片類似。21040是為PCI總線結構特別設計的。顯然目前不再生產這些芯片了，因為Intel購買了DEC的半導體部分，更傾向于使用他們自己的以太網卡芯片。

對基于此芯片的網卡可以在兩個驅動程序之間進行選擇。上面討論了DE425驅動程序，還有一個通用的21040“tulip”驅動程序。

警告：即使你的網卡是基于該芯片的，驅動程序也可能出問題。Davies寫道：

“無法保証‘tulip.c’或‘de4x5.c’可以驅動任一基于DC2114x的網卡，除非說明支持該網卡。為什么？？因為一個寄存器，通用寄存器（CSR12），(1)DC21140A中的CSR12是可編程的，而各個銷售商的實現各不相同；(2)DC21142/3中的CSR12現在是個SIA控制寄存器（按照DC21041的方式）。唯一的希望是SROM可以編碼來幫助構建驅動程序。但這并非可以確保的解決方案，因為有些銷售商（如SMC的9332網卡）并未遵從Digital Semiconductor推荐的SROM編程格式。”

從非技術的角度上說，這就意味著你無法確定一個未知的采用DC2114x芯片的網卡可以使用Linux驅動程序，所以在購買網卡之前，最好確定一下可以退貨。

在很多近期的SMC EtherPower網卡中也發現21041芯片被用來代替21040。21140可以支持100Base-T，并使用21040芯片的Linux驅動程序。如果要對非DEC的網卡使用David的`de4x5`驅動程序，請先看一下

README.de4x5里的細節。

果有問題的話，請試著從Donald的ftp/WWW站點獲取最新的驅動程序版本。

Tulip Driver <http://www.scyld.com/network>

那里還有一個各種使用21040芯片的網卡/銷售商的（非完全）列表。

4.17 Farallon

Farallon銷售EtherWave適配器和收發器。其設備可以把多個10baseT設備構成菊花鏈。

4.17.1 Farallon Etherwave

狀態：支持，驅動程序名稱：3c509

據説這是一個包含EtherWave收發器的3c509兼容網卡。可以成功地使用目前的Linux的3c509驅動程序。對于一般的應用，該網卡過于昂貴了，但在某些特殊情況下可以使用它。小Hub的價格在$125以上，Etherwave還要在板卡的價格之上增加$75-$100——如果拉一根線太少的話就物有所值，如果是缺少兩個網絡就不值得這么做了。

4.17.2 Farallon PCI 593

狀態：支持，驅動程序名稱：de4x5, tulip

據説該網卡可以被de4x5驅動程序檢測到。

4.18 Fujitsu

與許多網絡芯片制造商不同，Fujitsu還制造和銷售一些基于他們自己芯片的網卡。

4.18.1 Fujitsu FMV-181/182/183/184

狀態：支持，驅動程序名稱：fmv18x

從驅動程序來看，這些網卡就是Fujitsu的MB86965網卡的簡單改進，這使它們與Allied Telesis的AT1700網卡很相似。

4.19 Hewlett Packard

272**網卡使用可編程I/O，與NE*000板卡類似，但是數據傳輸端口可以在不訪問的時候“關閉”，以避免自動探測驅動程序時的麻煩。

感謝Glenn Talbott幫助澄清了本節有關HP硬件版本號的疑惑。

4.19.1 HP Night Director+ 10/100

狀態：支持，驅動程序名稱：pcnet32

看來這些網卡使用的是AMD 79C972芯片。

4.19.2 27245A

狀態：支持，驅動程序名稱：hp (+8390)

是基于8比特8390的10BaseT網卡，出于所有8比特的緣故不推荐使用。几年前該網卡被重新設計為高度集成的，引起了一些只影響測試程序的初始化時的變化，對LAN驅動程序沒有影響。（新的網卡在切換進入和退出環回模式后并不立刻處于“准備好”的狀態。）

如果你想把該驅動程序作為一個可載入的模塊，應該閱讀9.2 (把以太網驅動程序作為模塊使用)以了解有關模塊的信息。

4.19.3 HP EtherTwist, PC Lan+ (27247, 27252A)

狀態：支持，驅動程序名稱：hp+ (+8390)

HP的PC Lan+與標准的HP PC Lan網卡不同。從v1.1.x開發時其驅動程序就被加入到標准內核的驅動程序列表中。它可以象ne2000一樣工作在PIO模式，也可以象wd8013一樣工作在共享內存模式。

47B是基于16比特8390的10BaseT w/AUI，而52A是基于16比特8390的ThinLAN w/AUI。這些網卡用于收發數據包緩沖的板上RAM為32K，而不是一般的16KB，而且都提供自動識別LAN連接器的功能。

如果你想把該驅動程序作為一個可載入的模塊，應該閱讀9.2 (把以太網驅動程序作為模塊使用)以了解有關模塊的信息。

4.19.4 HP-J2405A

狀態：支持，驅動程序名稱：lance

這些網卡比27247/27252A便宜而且稍微快一點兒，但少了一些特性，如AUI、ThinLAN連接與啓動PROM插座。這是一個相當通用的LANCE設計，但設計時的一個不起眼的決定使它與通用的“NE2100”驅動程序不兼容。感謝HP的Glenn Talbott提供的資料，使驅動程序能夠包含對它的特別支持（包括從板上讀取DMA通道）。

更多有關基于LANCE的網卡信息可以在7.7 (Notes on AMD...)中找到。

4.19.5 HP-Vectra On Board Ethernet

狀態：支持，驅動程序名稱：lance

HP-Vectra在主板上有一個AMD的PCnet芯片。 DMA選擇和芯片號碼信息可以在4.4.1 (AMD LANCE)找到。

更多有關基于AMD LANCE的以太網卡的技術信息可以在7.7 (Notes on AMD...)找到。

4.19.6 HP 10/100 VG Any Lan Cards (27248B, J2573, J2577, J2585, J970, J973)

狀態：支持，驅動程序名稱：hp100

該驅動程序還支持某些Compex VG產品。由于驅動程序支持ISA、EISA和PCI網卡，所以在對內核源碼運行`make config`時可以在ISA網卡下找到它。

4.19.7 HP NetServer 10/100TX PCI (D5013A)

狀態：支持，驅動程序名稱：eepro100

然這只是重新標記過的Intel EtherExpress Pro 10/100B網卡。參見Intel一節以了解更多信息。

4.20 IBM / International Business Machines

4.20.1 IBM Thinkpad 300

狀態：支持，驅動程序名稱：znet

它與基于Intel的Zenith Z-note兼容。參見4.42.1 (Z-note)以了解更多信息。

據信下面這個站點有一個新版本Thinkpad的全面的有用資料數據庫。我還沒有親自去查看過。

Thinkpad-info `<http://peipa.essex.ac.uk/html/linux-thinkpad.html>`

對不便于使用WWW瀏覽器的讀者，可以試試`peipa.essex.ac.uk:/pub/tp750/`

4.20.2 IBM Credit Card Adaptor for Ethernet

狀態：半支持，驅動程序名稱：?（獨立發行）

該PCMCIA網卡可以在Linux下正常使用。同樣需要指出的是，筆記本電腦需要有支持PCMCIA的芯片組，而且要把支持PCMCIA的補丁加進標准內核。參見本文檔中的8.2 (PCMCIA支持)。

4.20.3 IBM 10/100 EtherJet PCI

狀態：支持，驅動程序名稱：eepro100

據說該網卡與Intel EtherExpress Pro 100驅動程序兼容。

4.20.4 IBM Token Ring

狀態：半支持，驅動程序名稱：ibmtr

要支持令牌環不僅僅需要寫一個設備驅動程序，還要為令牌環編寫源路由例程。編寫時花費時間最多的地方就是源路由。

開始的驅動程序開發是針對IBM的ISA和MCA令牌環網卡，并在MCA 16/4 Megabit令牌環板上進行過測試，但在其它基于Tropic的板上也可以工作。

4.21 ICL Ethernet Cards

4.21.1 ICL EtherTeam 16i/32

狀態：支持，驅動程序名稱：eth16i

驅動程序支持該網卡的ISA (16i)和EISA (32)兩個版本。它使用的Fujitsu的MB86965芯片也被用在at1700網卡上。

4.22 Intel Ethernet Cards

注意，各種Intel網卡的命名是亂七八糟的。如果有問題，檢查網卡上主要芯片的i8xxxx號碼，對PCI網卡使用/proc目錄下的PCI信息，然后再與這里列出的數字進行比較。

4.22.1 Ether Express

狀態：支持，驅動程序名稱：eexpress

該網卡使用intel的i82586芯片。驅動程序的早期版本（在v1.2內核里）是alpha-測試驅動程序，大多數人在使用時會遇到麻煩。v2.0內核里的驅動程序要好得多，雖然其源碼還被列為實驗性的，而且在較快的機器上問題更多。

在驅動程序源碼開頭的注釋里列出了一些與這些網卡有關的問題（并進行了修正！）。至少有一個用戶報告說有死鎖，使得把驅動程序里所有的outb替換為outb_p的工作放慢下來。同時還要檢查一下驅動程序報告的RAM緩存大小與Intel配置工具報告的是否一致。

4.22.2 Ether Express PRO/10 (PRO/10+)

狀態：支持，驅動程序名稱：eepro

Bao Chau Ha編寫了這些網卡的驅動程序，早在1.3.x內核中就包含了它。該驅動程序也可以使用在一些基于i82595芯片的Compaq內建以太網系統上。可能需要在適當的地方使用網卡自帶的配置工具禁用PnP支持。

4.22.3 Ether Express PRO/10 PCI (EISA)

狀態：半支持，驅動程序名稱：?（獨立發行）

有一個在缺省內核外獨立發行的PCI版本的驅動程序。這些網卡使用PLX9036 PCI接口芯片和Intel的i82596 LAN控制器芯片。如果你的網卡上有i82557芯片，那么你的網卡就不是該類型網卡，而是下面要討論的版本，需要EEPro100驅動程序。

可以從如下站點獲得PRO/10 PCI網卡的alpha驅動程序及使用指導：

EEPro10 Driver <http://www.ultranet.com/~stalba/eep10pci.html>

如果是EISA網卡，可能要改一下驅動程序在不同情況下（PCI與EISA）使用的檢測機制。

4.22.4 Ether Express PRO 10/100B

狀態：支持，驅動程序名稱：eepro100

注意，該驅動程序不支持老式的100A網卡。驅動程序里列出的芯片號碼是i82557/i82558。更新驅動程序或尋找支持，請訪問：

EEPro-100B Page <http://www.scyld.com/network>

4.23 Kingston

Kingston制造各種網卡，包括NE2000+、基于AMD PCnet的網卡和基于DEC tulip的網卡。大多數網卡都可以使用相應的驅動程序。參見*Kingston Web Page* <http://www.kingston.com>。

4.24 LinkSys

LinkSys制造少量不同的NE2000兼容網卡、一些正統的ISA網卡、一些即插即用的ISA網卡以及一些基于支持ne2000-PCI芯片組的ne2000-PCI兼容網卡。它們的型號太多，這里就不一一列出了。

LinkSys對Linux是友好的，擁有一個專門的Linux支持WWW主頁，甚至在某些產品的包裝盒上印有Linux字樣。請訪問：

`http://www.linksys.com/support/solution/nos/linux.htm`

4.24.1 LinkSys Etherfast 10/100 Cards.

狀態：支持，驅動程序名稱：tulip

注意，這些網卡有几種“修訂版”（即使用不同的芯片組），都使用一樣的網卡名。第一種使用DEC芯片組。第二種修訂版使用Lite-On PNIC 82c168 PCI網絡接口控制器，該網卡的第三種修訂版使用LinkSys 82c169 NIC芯片。對后兩種版本網卡的支持已經被合并到標准的tulip驅動程序里——按照當前所用驅動程序的版本，可能需要進行版本升級來支持它們。

更多的有關PNIC信息可在下面找到：

`http://www.scyld.com/linux/drivers/pnic.html`

有關這些網卡各種版本的進一步資料可以在上面提到過的LinkSys的WWW站點得到。

4.24.2 LinkSys Pocket Ethernet Adapter Plus (PEAEPP)

狀態：支持，驅動程序名稱：de620

據說這是DE-620兼容網卡，而且據報道可以正常使用DE-620的驅動程序。參見4.14.6 (DE-620)以了解更多信息。

4.24.3 LinkSys PCMCIA Adaptor

狀態：支持，驅動程序名稱：de650 (?)

有人猜想這是重新標記過的DE-650。參見4.14.7 (DE-650)以了解更多信息。

4.25 Microdyne (Eagle)

Eagle Technology（又稱Novell網卡）被賣給Microdyne。如果在這里的列表上找不到你的網卡，請檢查本文檔的Novell一節。雖然Microdyne現在不再積極銷售網卡，在他們的網站`ftp.mcdy.com`上還有一些與產品相關的資料。

4.25.1 Microdyne Exos 205T

狀態：半支持，驅動程序名稱：?

是另一款基于i82586的網卡。Dirk Niggemann `dirk-n@dircon.co.uk`編寫了一個自己認為屬于“pre-alpha”的驅動程序，希望有人進行測試。要了解細節請寫信給他。

4.26 Mylex

如果想詢問有關Mylex的問題，請通過如下號碼與他們聯系。

```
MYLEX CORPORATION, Fremont
Sales:  800-77-MYLEX, (510) 796-6100
FAX:    (510) 745-8016.
```

他們也有一個WEB站點： *Mylex WWW Site* `<http://www.mylex.com>`

4.26.1 Mylex LNE390A, LNE390B

狀態：支持，驅動程序名稱：lne390 (+8390)

這些是相當老的使用與wd80x3相似的共享內存方式的EISA網卡。在目前的2.1.x系列內核里有它們的驅動程序。應該確定共享內存地址被設置在1MB以下，或者在機器所安裝物理RAM的最高地址之上。

4.26.2 Mylex LNP101

狀態：支持，驅動程序名稱：de4x5, tulip

這是基于DEC的21040芯片的PCI網卡。可以選擇10BaseT、10Base2和10Base5輸出。LNP101網卡被証實可以使用通用的21040驅動程序。

參見有關21040芯片一節（4.16.4 (DEC 21040)）以了解更多信息。

4.26.3 Mylex LNP104

狀態：半支持，驅動程序名稱：de4x5, tulip

LNP104使用DEC的21050芯片來處理四個獨立的10BaseT端口。它應該可以使用最近的可以共享中斷的21040驅動程序，但（我覺察到）這一嘗試還沒有人報告使用過。

4.27 Novell Ethernet, NExxxx及其相關兼容卡。

前綴“NE”來自Novell Ethernet。Novell按照最便宜的NatSemi數據手冊設計并把制造權賣（扔？）給了Eagle，就是為了向市場提供價格合理的以太網卡。（目前被廣泛使用的NE2000網卡。）

4.27.1 NE1000, NE2000

狀態：支持，驅動程序名稱：ne (+8390)

ne2000現在是按照NatSemi的8390芯片進行基本設計的通用名稱。它們使用可編程I/O，而不是共享內存，使得安裝簡單，但性能有一點降低，而且帶來几個問題。有關NE2000網卡的一些普遍性問題列在2.5 (Problems with...)里。

有些NE2000兼容卡使用National Semiconductor的“AT/LANTic”83905芯片，提供與wd8013相似的共享內存模式和EEPROM軟件配置。共享內存模式比可編程I/O模式占用的CPU要少（也就是更有效）。

一般來說，把NE2000兼容卡的I/O地址放在`0x300`不是個好主意，因為几乎每一個設備驅動程序啓動時都探測這個地址。某些差勁的NE2000兼容卡在被探測到錯誤區域時處理不當會導致機器死鎖。同樣，`0x320`也不好，因為SCSI驅動程序會探測`0x330`。

Donald為所有ne2000網卡編寫了一個NE2000診斷程序（ne2k.c）。參見6.2 (診斷程序)以了解更多信息。

如果你想把該驅動程序作為一個可載入的模塊，應該閱讀9.2 (把以太網驅動程序作為模塊使用)以了解有關模塊的信息。

4.27.2 NE2000-PCI (RealTek/Winbond/Compex)

狀態：支持，驅動程序名稱：ne, ne2k-pci (+8390)

是的，無論相信與否，人們制造的PCI網卡是基于十多年前設計的ne2000接口。目前几乎所有這些網卡都是基于RealTek的8029芯片或Winbond的89c940芯片。顯然Compex、KTI、VIA和Netvin的網卡也使用這些芯片，只是PCI ID不同。

最新的v2.0內核支持對所有這些網卡的自動檢測與使用。（如果你使用v2.0.34之前的內核版本，把它升級以確保網卡被檢測到。）目前有兩個可選的驅動程序，一個是原始的ISA/PCI驅動程序`ne.c`，另一個是相對較新的只支持PCI的驅動程序`ne2k-pci.c`。

要使用原始的ISA/PCI驅動程序，在運行`make config`時必須對“Other ISA cards”選項回答“Y”，與ISA網卡使用同一個NE2000驅動程序。（這會讓你意識到這些網卡并不像PCNet-PCI或DEC 21040網卡一樣智能......）

較新的只支持PCI的驅動程序與ISA/PCI驅動程序的區別在于去掉了對所有老式8比特網卡的支持，而且在移動網卡上的數據時使用更大的數據塊，不再有老式ISA-NE2000網卡可靠操作所必需的中間暫停。這樣驅動程序變得小了一些，效率也高了一些，但在普通應用中這一區別并沒明顯到令人激動的地步。（如果希望得到最高效率/最低的CPU占用率，那么PCI-NE2000不過是個差勁的選擇。）驅動程序更新及更多信息可以在下面找到：

`http://www.scyld.com/linux/drivers/ne2k-pci.html`

如果你的NE2000 PCI網卡沒有被最新版本的驅動程序檢測到，請與`/usr/src/linux/MAINTAINERS`里列出的NE2000驅動程序維護者聯系，并附上`cat /proc/pci`和`dmesg`的輸出，以便把對該網卡的支持加到驅動程序里。

還要注意，有些網卡的制造商在明知道他們的網卡完全不同時（如PCNet-PCI或RealTek的8139），依然在產品包裝上寫著“NE2000 Compatible”。如果有疑問的話，按照本文檔查看一下主要芯片的編號。

4.27.3 NE-10/100

狀態：不支持。

些是基于National Semiconductor的DP83800和DP83840芯片的ISA 100Mbps網卡。目前沒有驅動程序支持它，也沒聽說有誰在編寫這樣的驅動程序。除了一個簡單的PDF文件，目前還沒有足以編寫驅動程序的有關該芯片的資料。

4.27.4 NE1500, NE2100

狀態：支持，驅動程序名稱：lance

這些網卡使用AMD的原始7990 LANCE芯片，可以使用Linux的lance驅動程序。較新的NE2100兼容卡使用AMD更新后的PCnet/ISA芯片。

某些老版本的lance驅動程序在使用autoIRQ從原始Novell/Eagle 7990網卡獲取中斷線時會遇到麻煩。希望現在已經被修正了。如果問題沒解決，就用LILO來指定IRQ，還有問題的話，請告訴我們。

DMA選擇和芯片號碼信息可以在4.4.1 (AMD LANCE)找到。

更多有關基于AMD LANCE的以太網卡的技術信息可以在7.7 (Notes on AMD...)找到。

4.27.5 NE/2 MCA

狀態：半支持，驅動程序名稱：ne2

若干公司生產了几種NE2000微通道網卡。v2.2內核里提供的ne2驅動程序可以檢測如下MCA網卡：Novell Ethernet Adapter NE/2、Compex ENET-16 MC/P和Arco Ethernet Adapter AE/2。

4.27.6 NE3200

狀態：不支持。

這種老式的EISA網卡把一片8MHz的80186與一片i82586混合使用。沒人為它編寫驅動程序，因為沒有該網卡的資料，而且也沒有對其驅動程序的真正需求。

4.27.7 NE3210

狀態：支持，驅動程序名稱：ne3210 (+8390)

這一EISA網卡與NE3200完全不同，它使用Nat Semi的8390芯片。其驅動程序可以在v2.2內核源碼樹中找到。確定共享內存地址被設置在1MB以下，或者在機器所安裝物理RAM的最高地址之上。

4.27.8 NE5500

狀態：支持，驅動程序名稱：pcnet32

這些不過是AMD PCnet-PCI網卡（’970A）芯片。更多有關基于LANCE/PCnet的網卡信息可以在4.4.1 (AMD LANCE)找到。

4.28 Proteon

4.28.1 Proteon P1370-EA

狀態：支持，驅動程序名稱：ne (+8390)

明顯這是一款NE2000兼容卡，在Linux下工作得挺好。

4.28.2 Proteon P1670-EA

狀態：支持，驅動程序名稱：de4x5, tulip

這是另一款基于DEC的Tulip芯片的PCI網卡。據說在Linux下工作得挺好。

參見有關21040芯片一節（4.16.4 (DEC 21040)）以了解更多信息。

4.29 Pure Data

4.29.1 PDUC8028, PDI8023

狀態：支持，驅動程序名稱：wd (+8390)

PureData的PDUC8028和PDI8023系列網卡差不多就是wd80x3的兼容網卡——在`wd.c`驅動程序里有特別的代碼來探測這些網卡。

4.30 Racal-Interlan

可以通過WWW站點`www.interlan.com`聯系Racal Interlan。我認為在過去他們曾被稱為MiCom-Interlan。

4.30.1 ES3210

狀態：半支持，驅動程序名稱：es3210

這是一款基于EISA 8390的共享內存網卡。有一個實驗性的驅動程序與v2.2內核一起發行，據說工作得挺好，但（至少）對早期版本網卡的EISA IRQ和共享內存地址的檢測有點問題。（這個問題并不僅限于Linux......）這樣就需要給驅動程序提供中斷和地址。例如，網卡為IRQ 5，共享內存在`0xd0000`，使用模塊化驅動程序，則在`/etc/conf.modules`里添上`options es3210 irq=5 mem=0xd0000`。或者使用編譯進內核的驅動程序，在啓動時提供`ether=5,0,0xd0000,eth0`。I/O基址是自動檢測的，所以在此使用一個零值。

4.30.2 NI5010

狀態：半支持，驅動程序名稱：ni5010

以前需要另外尋找這些老式8比特MiCom-Interlan網卡的驅動程序，現在它被作為一個實驗性的驅動程序與v2.2內核一起發行了。

4.30.3 NI5210

狀態：半支持，驅動程序名稱：ni52

網卡也是使用Intel的芯片。Michael Hipp編寫了一個驅動程序，作為“alpha”驅動程序被包含在標准內核里。Michael希望得到該網卡用戶的反饋。參見2.2 (Alpha驅動程序)以了解在Linux下使用alpha/測試版本以太網驅動程序的重要信息。

4.30.4 NI6510（不是EB）

狀態：半支持，驅動程序名稱：ni65

也有一個基于LANCE的NI6510網卡的驅動程序，而且也是Michael Hipp編寫的。它同樣也是“alpha”驅動程序。出于某些原因，該網卡與通用的LANCE驅動程序不兼容。參見2.2 (Alpha驅動程序)以了解在Linux下使用alpha/測試版本以太網驅動程序的重要信息。

4.30.5 EtherBlaster（又名NI6510EB）

狀態：支持，驅動程序名稱：lance

從1.3.23內核開始，通用的lance驅動程序里增加了對NI6510EB專有標識`0x52, 0x44`進行的檢查。但有人報告説不是所有NI6510EB網卡該標識都一樣，這會導致lance驅動程序檢測不出你的網卡。如果出現了這一問題，可以把探測（大約在lance.c的第322行）改為printk()，把你的網卡上的值打印出來，然后用它們來代替缺省的`0x52, 0x44`。

在使用lance驅動程序時，網卡可能運行在“高性能”模式下，而不是NI6510兼容模式。

4.31 RealTek

4.31.1 RealTek RTL8002/8012 (AT-Lan-Tec) Pocket adaptor

狀態：支持，驅動程序名稱：atp

這是一款AT-Lan-Tec與（可能的）其它一些供應商出售的通用低成本的OEM袖珍適配器。在標准內核里包含其驅動程序。注意，驅動程序源文件“atp.c”里包含了一些實質性的信息。

注意，對于早期版本的驅動程序，傳遞給`ifconfig`的設備名不是`eth0`，而是`atp0`。

4.31.2 RealTek 8009

狀態：支持，驅動程序名稱：ne (+8390)

這是一款ISA NE2000兼容網卡，據説使用Linux的NE2000驅動程序工作得挺好。程序`rset8009.exe`可以從RealTek的WWW站點`http://www.realtek.com.tw`獲取——或者用FTP從該站點下載。

4.31.3 RealTek 8019

狀態：支持，驅動程序名稱：ne (+8390)

這是上述網卡的即插即用版本。使用DOS軟件禁止PnP功能，啓用無跳線配置，把網卡的I/O地址和中斷設為可用值，然后就行了。（如果要把驅動程序用作模塊的話，不要忘記在`/etc/conf.modules`里添

加一個`io=0xNNN`選項。）程序`rset8019.exe`可以從RealTek的WWW站點`http://www.realtek.com.tw`獲取——或者用FTP從該站點下載。

4.31.4 RealTek 8029

狀態：支持，驅動程序名稱：ne, ne2k-pci (+8390)

這是一款NE2000兼容的PCI單芯片產品。現在各個銷售商都在銷售使用該芯片的網卡。參見4.27.2 (NE2000-PCI)以了解使用該網卡的有關信息。注意，這依然是附在PCI總線上的十多年前的老式設計。其性能并不比相應的ISA型號好多少。

4.31.5 RealTek 8129/8139

狀態：半支持，驅動程序名稱：rtl8139

另一款RealTek的PCI單芯片以太網產品。基于該芯片的網卡的驅動程序被包括在Linux的v2.0.34發行版本中。目前在回答是否要使用v2.2內核里的實驗性驅動程序時，依然要回答“Y”以獲得使用該驅動程序的權利。

Donald說基于該芯片的網卡的價格與PCI NE2000兼容網卡大致相同（在有些地方甚至會便宜13美元！），雖然8139的設計并非最好的10/100板子，還是要比PCI NE2000兼容網卡好。

2.4.x內核還有一個叫做`8139too`的驅動程序，是基于rtl8139的，但試圖處理一些人們報告的更普遍的問題，所以如果使用的是2.4內核，可以試一下。

4.32 Sager

4.32.1 Sager NP943

狀態：半支持，驅動程序名稱：3c501

這只不過是一款S.A. PROM前綴不同的3c501兼容網卡。我認為它跟原始的3c501一樣傷腦筋。其驅動程序檢查NP943 I.D.之后把它當成3c501來處理。參見4.1.1 (3Com 3c501)以了解為什么實際上不需要使用這些網卡。

4.33 Schneider & Koch

4.33.1 SK G16

狀態：支持，驅動程序名稱：sk_g16

其驅動程序被包含在v1.1內核中，是PJD Weichmann和SWS Bern編寫的。似乎SK G16與NI6510相似，都是基于LANCE芯片的第一版（7990）。再重復一次，似乎該網卡無法使用通用的LANCE驅動程序。

4.34 SEEQ

4.34.1 SEEQ 8005

狀態：支持，驅動程序名稱：seeq8005

驅動程序里只包含了很少的網卡信息，所以這里也沒有多少相關信息。如果有問題的話，最好發e-mail給源碼中列出的驅動程序作者。

4.35 SMC (Standard Microsystems Corp.)

在許多年前wd8003和wd8013還是主流產品時，Western Digital的以太網部門就被SMC收購了。在那之后，SMC繼續制造基于8390的ISA網卡（Elite16、Ultra、EtherEZ），并增加了几種PCI產品。

與SMC的聯系方法：

SMC / Standard Microsystems Corp., 80 Arkay Drive, Hauppage, New York, 11788, USA. Technical Support via phone: 800-992-4762 (USA) or 800-433-5345 (Canada) or 516-435-6250 (Other Countries). Literature requests: 800-SMC-4-YOU (USA) or 800-833-4-SMC (Canada) or 516-435-6255 (Other Countries). Technical Support via E-mail: `techsupt@ccmail.west.smc.com`. FTP Site: `ftp.smc.com`. WWW Site: *SMC* <`http://www.smc.com`>.

4.35.1 WD8003, SMC Elite

狀態：支持，驅動程序名稱：wd (+8390)

這些是網卡的8比特版本。8比特的8003比較便宜，但只在流量小的用途上是划算的。注意，某些沒有EEPROM的網卡（使用跳線的兼容卡，或非常非常古老的wd8003網卡）無法報告它們使用的中斷線。此時，如果使用auto-irq失敗，驅動程序就會靜悄悄地分配IRQ 5。你可以從SMC的FTP站點獲取設置/驅動程序磁盤。注意，某些較新的SMC的“SuperDisk”程序檢測不到那些確實古老的無EEPROM的網卡。文件`SMCDSK46.EXE`看來是個相當全面的方法。前面提到的檔案中的一個ASCII文本文件里給出了所有這些網卡的跳線設置。最新（最好？）的版本可以從`ftp.smc.com`得到。

這些基本上與它們的16比特對應網卡（WD8013 / SMC Elite16）是一樣，所以可以從下一節了解更多的信息。

4.35.2 WD8013, SMC Elite16

狀態：支持，驅動程序名稱：wd (+8390)

若干年后，SMC在設計中添加了更多的寄存器和一個EEPROM。（第一塊wd8003網卡已經面世十年了！）兼容卡一般使用的名稱為“8013”，而且通常采用無EEPROM（用跳線）的設計。近期型號的SMC網卡使用SMC的83c690芯片，代替了早期網卡上的原始Nat Semi的DP8390。共享內存設計使網卡的速度比PIO網卡快了一點，尤其是對較大的數據包。更重要的是，從驅動程序的角度來看，避免了8390可編程I/O模式的几個Bug，允許多個線程安全地訪問數據包緩存，而且在熱啓動探測時不會因為可編程I/O數據寄存器導致死機。

無EEPROM的網卡在無法讀取選定的中斷時會嘗試auto-irq，如果失敗的話，它會靜悄悄地分配IRQ 10。（8比特版本分配IRQ 5。）

板上內存大小不標准的網卡可以在啓動時指定內存大小（如果使用模塊的話，把它作為`/etc/conf.modules`中的一個選項）。8比特網卡的標准內存大小為8kB，16比特網卡為16kB。例如，老式的WD8003EBT網卡可以用跳線設置為32kB內存。要完全使用這些RAM，需要如下操作（對I/O=0x280和IRQ 9）：

```
LILO: linux ether=9,0x280,0xd0000,0xd8000,eth0
```

參見2.6 (8013問題)以了解某些常見問題。

果你想把該驅動程序作為一個可載入的模塊，應該閱讀9.2 (把以太網驅動程序作為模塊使用)以了解有關模塊的信息。

4.35.3 SMC Elite Ultra

狀態：支持，驅動程序名稱：smc-ultra (+8390)

該以太網卡是基于SMC的83c790芯片，比83c690多几個新特性。雖然有一種與老式SMC以太網卡相似的模式，它無法與老的WD80*3驅動程序完全兼容。但在這種模式下，它可以與其它8390驅動程序共享大部分代碼，而且操作比WD8013兼容網卡稍微快一點。

由于Ultra部分看起來象8013，所以Ultra探測應當在wd8013探測錯誤地識別一個Ultra網卡之前找到它。

Donald提到可以為Ultra的“Altego”模式編寫一個獨立的驅動程序，該模式允許把發送鏈接起來進行，其代價是接收緩存的利用效率會下降，但是也可能不下降。

總線控制SCSI主機適配器的用戶需要注意：在與Interactive UNIX一起發行的手冊里提到，SMC Ultra里的一個Bug會導致aha-154X主機適配器運行的SCSI磁盤數據損壞。這可能影響aha-154X兼容卡，如BusLogic板和AMI-FastDisk SCSI主機適配器。

SMC承認在Interactive和老式Windows NT驅動程序上會出現這個問題。它是早期版本的網卡存在的硬件沖突，可以通過驅動程序的設計解決。目前的Ultra驅動程序通過只在與網卡傳遞數據時才啓用共享內存避免了這個問題。確定你的內核版本至少是1.1.84以上，或者啓動時報告的驅動程序版本至少為`smc-ultra.c:v1.12`，否則就會出問題。

如果你想把該驅動程序作為一個可載入的模塊，應該閱讀9.2 (把以太網驅動程序作為模塊使用)以了解有關模塊的信息。

4.35.4 SMC Elite Ultra32 EISA

狀態：支持，驅動程序名稱：smc-ultra32 (+8390)

這個EISA網卡與其相應的ISA網卡有很多共同點。在v2.0和v2.2內核里都包含了一個可以使用（而且穩定）的驅動程序。感謝Leonard Zubkoff買了一些這樣的網卡，使得Linux里能夠增加對它們的支持。

4.35.5 SMC EtherEZ (8416)

狀態：支持，驅動程序名稱：smc-ultra (+8390)

該網卡使用SMC的83c795芯片并支持即插即用特性。它也有*SMC Ultra*兼容模式，可以使用Linux的Ultra驅動程序。要得到最佳效果，請使用SMC提供的程序（通過他們的FTP/WWW站點）禁用PnP，并把它配置為共享內存模式。參見上面有關Ultra驅動程序的說明。

對于v1.2內核，該網卡必須被配置為共享內存模式。但v2.0內核可以在共享內存或可編程I/O模式下使用該網卡。共享內存模式稍微快一點，而且對CPU資源的占用少一些。

4.35.6 SMC EtherPower PCI (8432)

狀態：支持，驅動程序名稱：de4x5, tulip

先聲明一點：EtherPower II是一款完全不同的網卡。往下看！這些網卡是基本的DEC 21040產品，即一個大芯片和若干收發器。Donald在開發通用21040驅動程序（又名`tulip.c`）時使用了一塊這樣的網卡。再一次感謝Duke Kamstra提供了開發時所用的網卡。

某些近期版本的這種網卡使用較新的DEC 21041芯片，在使用早期版本的tulip驅動程序時可能會有問題。如果遇到麻煩，請確定使用的是最新發布的驅動程序，它可能還沒有被包含在目前的内核源碼樹中。

參見4.16.4 (DEC 21040)以了解這些網卡的更多信息和當前驅動程序的狀況。

很明顯，EtherPower-II網卡的最新版本使用9432芯片。目前還不清楚它是否可以使用現在的驅動程序。同樣，如果無法確信的話，在購買網卡之前，先確定如果不能使用Linux驅動程序就可以退貨。

4.35.7 SMC EtherPower II PCI (9432)

狀態：半支持，驅動程序名稱：epic100

這些網卡是基于SMC 83c170芯片，與基于Tulip的網卡完全不同。在v2.0和v2.2内核里包含支持它的新驅動程序。更多的細節請訪問：

`http://www.scyld.com/linux/drivers/epic100.html`

4.35.8 SMC 1211TX 10/100

狀態：半支持，驅動程序名稱：rtl8139

顯然SMC已經不再是如Ultra和EPIC那樣提供網卡的同一家公司。芯片設計部分現在叫做SMSC，而SMC的名稱是與低端OEM板聯系在一起的，就像這塊板一樣——帶一塊修改過的EEPROM的RealTek 8139。

4.35.9 SMC 3008

狀態：不支持。

這些8比特網卡是基于Fujitsu MB86950的，它是Linux at1700驅動程序所用MB86965的古老版本。Russ說可以通過閱讀at1700.c代碼和他自己的Tiara網卡DOS驅動程序包（tiara.asm）來編寫它的驅動程序。這些網卡不是很常見。

4.35.10 SMC 3016

狀態：不支持。

這些是16比特的I/O映射8390網卡，與通用NE2000網卡很相似。如果你能從SMC得到説明資料，那么移植NE2000驅動程序將相當簡單。這些網卡不是很常見。

4.35.11 SMC-9000 / SMC 91c92/4

狀態：支持，驅動程序名稱：smc9194

SMC9000是基于91c92芯片的VLB網卡。91c92也出現在若干其它品牌網卡上，但相當少見。

4.35.12 SMC 91c100

狀態：半支持，驅動程序名稱：smc9194

說SMC 91c92驅動程序支持該款基于100Base-T芯片的網卡，但目前還未得到証實。

4.36 Texas Instruments

4.36.1 ThunderLAN

狀態：支持，驅動程序名稱：tlan

該驅動程序覆蓋了許多Compaq的內建以太網設備，包括NetFlex和Netelligent系列。它還支持Olicom的2183、2185、2325和2326產品。

4.37 Thomas Conrad

4.37.1 Thomas Conrad TC-5048

這是基于DEC的21040芯片的另一款PCI網卡。

參見有關21040芯片一節（4.16.4 (DEC 21040)）以了解更多信息。

4.38 VIA

你可能沒見過VIA網卡，但VIA制造的几種網絡芯片被用在其它廠家的以太網卡上。他們有一個WWW站點：

`http://www.via.com.tw/`

4.38.1 VIA 86C926 Amazon

狀態：支持，驅動程序名稱：ne, ne2k-pci (+8390)

該控制器芯片是VIA提供的PCI-NE2000。你可以選擇ISA/PCI驅動程序`ne.c`或者只支持PCI的驅動程序

```
<tt/ne2k-pci.c/
```

。

參見PCI-NE2000一節以了解更多信息。

4.38.2 VIA 86C100A Rhine II (and 3043 Rhine I)

Status 支持，驅動程序名稱：via-rhine

這一相當新的驅動程序可以在目前的2.0和2.1內核里找到。它是在86C926 NE2000芯片上的改進，支持總線控制權傳遞，但嚴格的緩存32比特字節對齊的要求限制了從中得到的好處。要了解更多細節和驅動程序更新，參見：

`http://www.scyld.com/linux/drivers/via-rhine.html`

4.39 Western Digital

請參見4.35 (SMC)以了解有關SMC網卡的信息。（SMC在多年前就收購了Western Digital的網卡部門。）

4.40 Winbond

Winbond實際上并不制造和向大衆銷售完整的網卡——他們生產單片以太網芯片，其它公司購買并在PCI板上印上自己的名字，然后通過零售店銷售。可以從下面得到一些設置程序和技術支持：

`http://www.winbond.com.tw`

4.40.1 Winbond 89c840

狀態：半支持，驅動程序名稱：winbond-840

該芯片被描述為“NE2000網卡和Tulip兼容網卡雜交産生的突變體”——參見驅動程序的注釋以了解更多 細節。該驅動程序還沒有與内核一起發行，因為它還處于測試階段（從1998年9月開始）。可以從下面獲取驅動程序：

`http://www.scyld.com/linux/drivers/test/winbond-840.c`

4.40.2 Winbond 89c904, 89c905, 89c906

狀態：支持，驅動程序名稱：ne (+8390)

這些是Winbond的ISA 10Mbps ne2000兼容以太網芯片。設置程序可以從Winbond站點獲得。

4.40.3 Winbond 89c940

狀態：支持，驅動程序名稱：ne, ne2k-pci (+8390)

該芯片是很多制造商出售的低價PCI ne2000網卡上最常見的兩個芯片之一。注意，這依然是附在PCI總線上的十多年前的老式設計。其性能并不比相應的ISA型號好多少。

4.41 Xircom

在很長一段時間里，除非你簽署文件放棄你的生命，Xircom不愿意發布編寫驅動程序所需的編程信息。顯然有足夠多的Linux用戶不斷糾纏他們，要得到驅動程序支持（Xircom聲稱支持所有通用網絡操作系統......），所以他們改變了政策，無需再簽署不公開協議，允許公開有關資料。有些人説他們將要公開SCO驅動程序的源碼，另一些被告知他們不再提供象早期PE型號那樣“被廢棄”的産品資料。如果你感興趣并想自己了解一下，可以聯系Xircom：1-800-874-7875, 1-800-438-4526 或 +1-818-878-7600。

4.41.1 Xircom PE1, PE2, PE3-10B*

狀態：不支持。

不要抱太大希望，但如果你有一個這樣的并口適配器，可以在DOS仿真器下使用Xircom提供的DOS驅動程序使用它。你需要允許DOSEMU訪問并口，可能還要和SIG（DOSEMU的傻瓜中斷生成器）一起使用。

4.41.2 Xircom PCMCIA Cards

狀態：半支持，驅動程序名稱：????

在David Hinds的PCMCIA軟件包里有一些Xircom的PCMCIA網卡驅動程序。查看一下以了解最新信息。

4.42 Zenith

4.42.1 Z-Note

狀態：支持，驅動程序名稱：znet

內建的Z-Note網絡適配器是基于使用兩個DMA通道的Intel i82593。在目前的內核版本里有一個（alpha?）驅動程序。與所有筆記本電腦和袖珍適配器一樣，在運行`make config`時它被放在“Pocket and portable adaptors”部分。同時還要注意，IBM ThinkPad 300與Z-Note兼容。

4.43 Znyx

4.43.1 Znyx ZX342 (DEC 21040 based)

狀態：支持，驅動程序名稱：de4x5, tulip

在使用基于該芯片的網卡時可以在兩個驅動程序中進行選擇。David編寫了DE425驅動程序，Donald編寫了通用的21040驅動程序。

注意，在1.1.91，David增加了一個編譯選項，允許非DEC網卡（如Znyx網卡）使用該驅動程序。請閱讀`README.de4x5`以了解其中細節。

參見4.16.4 (DEC 21040)以了解這些網卡的更多信息和當前驅動程序的狀況。

4.44 識別未知網卡

好吧，假設你叔叔的侄子的鄰居的朋友有一個兄弟，在他給兒子的寵物大老鼠做窩的AT箱子里找到了一塊老式ISA以太網卡。不知什么原因這塊網卡最后落到你的手里，想在Linux下使用，但沒人知道這塊網卡是什么，也沒有任何資料。

首先，尋找可以作為線索的任何明顯的型號編碼。如果型號里包含2000，那么很有可能是塊NE2000兼容網卡。上面有8003和8013的網卡可能是Western/Digital的WD80x3網卡，或者是SMC Elite網卡，或者它們的兼容網卡。

4.44.1 識別網絡接口控制器

尋找網卡上最大的芯片。那就是網絡控制器（NIC），大多數都可以從部分編號上識別出來。如果你知道網卡上是哪一種NIC，下文就能幫你弄清是哪一種網卡。

可能最常見的NIC還是National Semiconductor的DP8390，又名NS32490，又名DP83901，又名DP83902，又名DP83905，又名DP83907。這些不過是National生產的一部分芯片！其它一些公司，如Winbond和UMC，生產DP8390和DP83905兼容部件，如Winbond的89c904（與DP83905兼容）和UMC的9090。如果網卡由8390構成，那么它可能就是一個ne1000或ne2000兼容網卡。其次常用的基于8390的網卡是wd80x3網卡及其兼容卡。使用DP83905的網卡可以被認為是ne2000或wd8013。新版本的通用wd80x3和SMC Elite網卡在原來DP8390的位置上用的是83c690。SMC Ultra網卡采用83c790，使用與wd80x3網卡稍有區別的驅動程序。SMC EtherEZ網卡采用83c795，使用的驅動程序與SMC Ultra網卡一樣。所有基于某種8390或8390兼容芯片的BNC網卡都有一片緊靠BNC連接器的16針DIP芯片8392（或83c692，又或???392）。

另一個在老式網卡上找到的常見NIC是Intel的i82586。使用該NIC的網卡包括3c505、3c507、3c523、Intel的EtherExpress-ISA、Microdyne的Exos-205T和Racal-Interlan的NI5210。

原始AMD LANCE NIC的編號為AM7990，較新的版本包括79c960、79c961、79c965、79c970和79c974。上面的大多數網卡都可以使用Linux的LANCE驅動程序，有個例外是使用自己驅動程序的老式Racal-Interlan的NI6510網卡。

使用DEC的21040、21041、21140或類似的編號NIC的較新的PCI網卡可以使用Linux的tulip或de4x5驅動程序。

其它有一個印著RTL8029、89C940或86C926的大芯片的PCI網卡是ne2000兼容網卡，v2.0以上版本Linux里的ne驅動程序在啓動時可以自動檢測這些網卡。

4.44.2 識別以太網地址

每一個以太網卡都有一個唯一的6字節網卡地址。同一個制造商生產的每一塊網卡的該地址的前三個字節都是一樣的。例如，所有的SMC網卡都以`00:00:c0`開始。制造商分配的后三個字節對于自己生產的每一塊網卡都是唯一的。

如果你的網卡上有個標簽給出了網卡地址的全部六個字節，就可以通過前三個字節查到它的銷售商。但一般PROM插座上貼的標簽上只打印了后三個字節，沒什么用。

你可以從RFC-1340上找到分配給銷售商的地址。顯然有很多地方可以找到最新的地址列表。通過WWW或FTP查找`EtherNet-codes`或`Ethernet-codes`就可以找到一些資料。

4.44.3 通過FCC ID號碼識別網卡

作為網卡通常在銷售給用戶之前必須通過的驗証過程的一部分，網卡要被FCC測試，并把由此獲得的FCC ID印在網卡上。例如，某塊網卡上有`FCC ID: J158013EWC`——那么該網卡就是SMC/WD8013-EWC。`www.driverguide.com`和`drdriver.com`一類的某些WEB站點使用FCC ID列表來幫助ID號不明顯的情況。

4.44.4 使用未知網卡的技巧

如果還無法確定是哪一種網卡，但至少縮小了可能的范圍，那么你可以構建一個包括全部可能驅動程序的內核，然后看看其中的哪一個可以在啓動時自動檢測到網卡。

如果內核沒有檢測到網卡，那么就是網卡沒有被配置到一個驅動程序尋找網卡時探測的地址上。此時，需要從當地的Linux的FTP站點下載`scanport.tar.gz`，看看它能否定位出你的網卡跳線所設的地址

。它掃描從`0x100`到`0x3ff`的ISA I/O地址以尋找`/proc/ioports`里沒有注冊的設備。如果它在某個特定地址發現了一個未知設備，那么就可以用一個啓動參數`ether=`把它明確設定為以太網卡探測的地址。

果你設法檢測到了網卡，那么一般可以依次改變未知跳線來了解跳線的作用，并查看被檢測到的網卡所用I/O基址和中斷。中斷設置一般可以通過查看跳線在網卡背面所焊接的位置來得到。從網卡有金屬插座的一端數背面的“金手指”，相應于金手指4、21、22、23、24、25、34、35、36、37、38分別是中斷9、7、6、5、4、3、10、11、12、15、14。8比特網卡只用到金手指31。

看起來沒什么用的跳線一般是用來選擇可選啓動ROM的內存地址。靠近BNC或RJ-45或AUI的其它跳線一般是用來選擇輸出媒介的。一般它們也在印著YCL、Valor或Fil-Mag的“黑盒子”變壓器附近。

如下URL收集了很多各種網卡的跳線設置：

Ethercard Settings `<http://www.slug.org.au/NIC/>`

4.45 非以太網設備的驅動程序

在Linux源碼里有若干在網絡程序看來是象以太網設備的其它驅動程序，但實際上不是以太網設備。為了本文檔的完整性，把它們簡要地列在這里。

`dummy.c`——該驅動程序的目的是提供一個設備來指出路由，但并不實際傳輸數據。

`eql.c`——負荷均衡，控制多個設備（一般是modem）并在其中平衡發送負載，對網絡程序表現為單個設備。

`ibmtr.c`——IBM的令牌環，并非真正的以太網。令牌環要求源路由以及其它一些麻煩事。

`loopback.c`——環回設備，你的機器發出的所有數據包的目的地還是你自己的機器。從本質上說只是把數據包從發送隊列移到接收隊列。

`pi2.c`——Ottawa業余無線電愛好者俱樂部PI和PI2的接口。

`plip.c`——并行線Internet協議，允許兩台計算機通過并口的端到端方式連接互相發送數據包。

`ppp.c`——端到端協議（PPP，RFC1331），在端到端的連接（一般為modem）上傳輸多協議數據幀。

`slip.c`——串行線Internet協議，允許兩台計算機通過串口（一般通過modem）的端到端方式連接互相發送數據包。

`tunnel.c`——提供一個IP通道，可以使網絡數據流透明地穿過子網。

`wavelan.c`——由Intel的82586協處理器控制的類以太網無線電收發器。82586也被用在諸如Intel EtherExpress的其它以太網卡上。

5 電纜、同軸電纜、雙絞線

如果你是從頭開始建網絡，那么就必須決定是使用以太網細纜（使用BNC連接器的RG58同軸電纜）或10baseT（使用8線“電話”連接器RJ-45的電話線樣式的雙絞線電纜）。使用N連接器的RG-5電纜的老式以太網粗纜已經被廢棄，現在很少見到了。

參見1.4 (網卡應當支持的線纜類型)以大致了解一下線纜類型。還要注意，*comp.dcom.lans.ethernet*里的FAQ包括了大量有關電纜方面的有用信息。FTP到rtfm.mit.edu下的`/pub/usenet-by-hierarchy/`以查看該新聞組的FAQ。

5.1 以太網細纜(thinnet)

以太網細纜相當便宜。如果自己制作電纜，實心的RG58A是$0.27/m，分股的RG58AU是$0.45/m。彎頭的BNC連接器每個< $2，其它各種器件也都挺便宜。重要的是在每段電纜的末端需要裝一個50歐姆的終結器，所以每條電纜的預算大約是$2。還有一個關鍵問題是電纜中不能有“殘樁”——“T”連接器必須直接連到以太網卡上。

使用以太網細纜有兩個主要缺點。首先它限制為10Mb/sec——100Mb/sec要求使用雙絞線。其次，如果你是用一個大環路把機器連接起來，而某些笨蛋把電纜從自己的T形接頭上拿開破壞了環路，整個網絡就無法使用了，因為出現了一個無限大的阻抗（電路開路）而不是要求的50歐姆終結。注意，只要不把電纜從T形接頭上移開，而是把T形接頭從網卡上取下來，整個子網就不會被破壞。當然，從機器上拔出T形接頭會干擾機器的工作。8-) 如果只是兩台機器的小網絡，還是需要T形接頭和50歐姆終結器——不能用直接用電纜把它們連起來！

有一些有趣的電纜系統看起來象用一根導線連到網卡上，但實際上是有兩根并排的導線包在外面的套子里，其截面為橢圓形。在環路的轉折點，用拼接的BNC連接器連到網卡上。所以它等同于用兩根電纜和一個BNC的T形接頭，但在這種情況下，用戶不可能從T形接頭的一邊移走電纜來干擾網絡。

5.2 雙絞線

雙絞線網絡需要主動式的Hub，每個Hub起碼要$50，未加工的電纜費用實際上比細纜要高。你可以不理會那些聲稱可以使用已有電話線的說法，因為這種安裝情況是很少見的。

另一方面，所有100Mb/sec以太網提議都使用雙絞線，而且絕大多數新的商業安裝使用的也是雙絞線。此外，Russ Nelson進一步提出“新的布線應該使用5類線。使用其它的電纜都是在浪費安裝者的時間，因為任何一種100Base都將要求使用5類線。”

如果只是連接兩台機器，可以通過交換接收與發送線（1-2和3-6）來避免使用Hub。

如果你拿著RJ-45連接頭沖著自己（就象你要把它插到自己嘴里），鎖扣朝上，那么從左到右各插腳的編號依次是1到8。各插腳的用途如下：

```
插腳編號          用途
----------                      ----------
1                 輸出數據 (+)
2                 輸出數據 (-)
3                 輸入數據 (+)
4                 保留為電話使用
5                 保留為電話使用
6                 輸入數據 (-)
7                 保留為電話使用
8                 保留為電話使用
```

如果你想自己制作一條電纜，請認真閱讀下文。一對差分信號必須使用同一條雙絞線，以滿足UTP電纜所要求的最小阻抗/損耗。如果看一下上面的表格，就知道1+2和3+6是兩對差分信號。不是1+3和2+6！！！！在10MHz，長度較短的情況下，出這樣的錯還可以工作，但僅限于線路長度較短的情況下。無法想象在100MHz時出錯會怎么樣。

對于末端為“A”和“B”的普通接插線，你要做一個插腳到插腳的映射，輸入和輸出分別使用一對雙絞線（出于阻抗的原因）。這就意味著1A連到1B，2A連到2B，3A連到3B，6A連到6B。連接1A-1B和2A-2B的必須是一條雙絞線。同樣，連接3A-3B和6A-6B的必須是另一條雙絞線。

如果目前沒有Hub，想做一條“空電纜”，那么要做的就是把“B”的輸出作為“A”的輸入，把“A”的輸出作為“B”的輸入，同時不改變它們的極性。也就是說，把1A連到3B（A的輸出+到B的輸入+），把2A連到6B（A的輸出-到B的輸入-）。這兩根線必須是一條雙絞線。它們把網卡/插腳“A”作為輸出傳送，把網卡/插腳“B”作為輸入傳送。然后把3A連到1B（A的輸入+到B的輸出+），把6A連到2B（A的輸入-到B的輸出-）。這兩根線也必須是一條雙絞線。它們把網卡/插腳“A”作為輸入傳送，把網卡/插腳“B”作為輸出傳送。

因此，如果想使用普通接插線，需要砍掉它的一端，交換接收和發送雙絞線的位置后插入新的插頭，把它壓緊，就得到一條“空”電纜了。一點也不復雜。只不過是把一塊網卡的發送信號送到第二塊網卡的接收端，如法炮制就可以了。

注意，在10BaseT被批准為標准之前，就存在其它使用RJ-45連接頭的網絡格式和上述的連線安排。例如SynOptics的LattisNet和AT&T的StarLAN。在某些情況下，（如使用早期的3C503網卡）可以設置網卡上的跳線使網卡能夠與不同類型的Hub連接，但絕大多數情況下，為這些老式類型網絡設計的網卡無法工作在標准10BaseT網絡/Hub下。（注意，如果網卡還有一個AUI端口，那么肯定可以把AUI和10BaseT收發器混合在一起使用。）

5.3 以太網粗纜

以太網粗纜几乎完全被廢棄了，一般只用在與一個已有網絡保持兼容。可以放松規定并使用被動的$3的N-to-BNC連接器把大量以太網粗纜和細纜連接起來，這通常是擴展已有以太網粗纜的最佳解決方案。對這種情況一個正確（但代價昂貴）的方案是使用轉發器。

6 軟件設置與網卡診斷

大多數情況下，如果用軟件進行配置并把配置保存在EEPROM里，一般需要重新啓動DOS，使用銷售商提供的DOS程序來設置網卡的中斷IRQ、I/O、內存地址和其它東西。此外，希望這是只要設置一次的東西。如果沒有網卡的DOS軟件，試著在網卡制造商的WWW站點上找找。如果不知道站點名稱，可以猜一下，如“www.my_vendor.com”，其中“my_vendor”是網卡制造商的名稱。這種方法對SMC、3Com和許多許多其它制造商都有效。

有些網卡有Linux版本的配置工具，并在本文中列出。Donald編寫了几個運行在Linux下的小的網卡診斷程序，其中大部分都是他在編寫驅動程序時使用的診錯工具。不要指望有花哨的菜單驅動界面。在使用絕大部分工具時，你需要閱讀源碼。即使你所用的網卡沒有相應的診斷工具，你還是可以通過`cat /proc/net/dev`獲得一些信息——假設至少在啓動時你的網卡被檢測到了。

隨便哪種情況下，都得在啓動時運行這些程序（允許對端口進行I/O操作），而且可能在這么做之前要先用`ifconfig eth0 down`關閉以太網卡。

6.1 以太網卡的配置程序

6.1.1 WD80x3網卡

對于使用wd80x3網卡的人，在Linux的FTP站點上可以找到的`wdsetup-0.6a.tar.gz`里有一個程序`wdsetup`。它的維護不是很好，有一段時間沒有更新了。如果能用，那么很好；如果不行，那么只好使用應當和網卡一起拿到的DOS版本程序。如果沒有這樣的DOS版本程序，還是可以很高興地告訴你，從SMC的FTP站點上可以獲取設置/驅動程序磁盤。當然，你必須有一個帶EEPROM的網卡才能使用該工具。很老很老的wd8003網卡和一些wd8013兼容網卡使用跳線來設置網卡。

6.1.2 Digital/DEC網卡

Digital的EtherWorks 3網卡可以用類似于DOS程序`NICSETUP.EXE`的方式進行配置。David C. Davies編寫了該工具以及和驅動程序在一起的其它EtherWorks 3工具。查看一下你本地的Linux FTP站點下的目錄`/pub/linux/system/Network/management`，尋找名為`ewrk3tools-X.XX.tar.gz`的文件。

6.1.3 NE2000+或AT/LANTIC網卡

有些Nat Semi的DP83905產品（如AT/LANTIC和NE2000+）是可以用軟件配置的。（注意，這些網卡也可以仿真wd8013網卡！）你可以從Donald的FTP服務器`www.scyld.com`下載文件`/pub/linux/setup/atlantic.c`來配置此網卡。另外，Kingston的DP83905網卡配置程序看起來可以用在所有網卡上，因為在允許使用網卡前，它們并不檢查銷售商指定的地址。按照如下URL： *Kingston Software* `<http://www.kingston.com/download/etherx/etherx.htm>` 下載`20XX12.EXE`和`INFOSET.EXE`。

在配置NE2000+網卡時要格外小心，因為設定的值有錯誤會帶來麻煩。一個典型的錯誤是啓用EEPROM里的啓動ROM（即使沒有安裝ROM）會和VGA卡發生沖突。其結果就是開機后屏幕沒有顯示，計算機只是發出一些叫聲。

通常如下操作就可以解決問題：拔出機器里的卡，然后啓動進入CMOS設置。把“Display Adapter”改為“Not Installed”并把缺省的啓動驅動器改為“A:”（軟盤驅動器）。同時把“Wait for F1 if any Error”改為“Disabled”。這樣，計算機就無需用戶干預可以啓動了。現在創建一個可啓動的DOS軟盤（“format a: /s /u”），把上述檔案文件`20XX12.EXE`里的程序`default.exe`拷貝到軟盤上。再鍵入`echo default > a:autoexec.bat`，這樣在用這張軟盤啓動時，該程序會自動運行，把網卡設置為正常的缺省值。關閉機器，重新安裝ne2000+網卡，插入新的啓動軟盤，再開機。可能機器還會發出叫聲，但最后你會看到機器從軟盤啓動時軟驅的燈亮了。等一兩分鐘，軟驅會停下來，表示`default.exe`程序運行結束，再關閉機器電源。再次啓動時，可能你的顯式器就又工作正常了，這時就可以把CMOS設置改回去，并把網卡EEPROM設置改回到希望的值。

注意，如果手頭沒有DOS，你可以用一張自動運行Donald的`atlantic`程序（使用正確的命令行開關）的Linux啓動軟盤來完成上面的工作，代替自動運行`default.exe`程序的DOS啓動軟盤。

6.1.4 3Com網卡

3Com的Etherlink III系列網卡（如3c5x9）可以用Donald的另一個配置工具進行配置。你可以從Donald的FTP服務器`www.scyld.com`上獲取文件`/pub/linux/setup/3c5x9setup.c`來配置這些網卡。（注意，DOS下的3c5x9B配置工具可能有一些屬于Etherlink III的新“B”系列的額外選項。）

6.2 以太網卡的診斷程序

Donald編寫的診斷程序都可以從以下URL獲取。

Ethercard Diagnostics `<http://www.scyld.com/network>`

Allied Telesis AT1700——`at1700.c`

Cabletron E21XX——`e21.c`

HP PCLAN+——`hp+.c`

Intel EtherExpress——`eexpress.c`

PCI NE2000網卡——`ne2k-pci-diag.c`

ISA NE2000網卡——`ne2k.c`

RealTek (ATP)袖珍適配器——`atp-diag.c`

所有其它網卡——試著用`cat /proc/net/dev`和`dmesg`來看看內核里有關所說網卡的有用信息。

7 技術信息

對于那些想了解更多有關網卡如何工作、或如何使用現有驅動程序，以及試圖為目前不支持的網卡編寫自己的驅動程序的人來說，這些信息應該會有用。如果你沒有這種想法，那么最好跳過這一節。

7.1 可編程I/O、共享內存與DMA

如果已經可以發送接收背靠背數據包，就無法把更多的數據放到網絡上。每一個現代以太網卡都可以接收背靠背數據包。Linux的DP8390驅動程序（wd80x3、SMC-Ultra、3c503、ne2000，等等）非常接近于發送背靠背數據包（依賴于當前的中斷延遲），3c509和AT1500的硬件在自動發送背靠背數據包上沒有一點問題。

7.1.1 可編程I/O（如NE2000、3c509）

優點：沒有使用任何受限制的系統資源，只用了若干I/O寄存器，而且沒有16M的限制。

缺點：一般傳輸速率較慢，CPU需要等待，几乎不可能訪問交叉的數據包。

7.1.2 共享內存（如WD80x3、SMC-Ultra、3c503）

優點：簡單，比可編程I/O速度快，允許隨機訪問數據包。在可能的情況下，Linux驅動程序在從網卡復制出接收的IP數據包時計算其校驗和，從而比相應的PIO網卡進一步減少了對CPU的占用。

缺點：占用內存空間（對DOS用戶來說是個大問題，在Linux下沒有問題），依然要占用CPU。

7.1.3 從屬（普通）的直接內存存取（Linux下沒有這種情況！）

優點：在實際數據傳遞過程中不占用CPU。

缺點：檢查邊界條件、分配相鄰的緩存和DMA寄存器編程使該方法成為最慢的技術。它還占用了一個珍貴的DMA通道，并要求對齊的低端內存緩存。

7.1.4 總線控制的直接內存存取（如LANCE、DEC 21040）

優點：在數據傳輸過程中不占用CPU，可以把緩存串起來，CPU時間很少或沒有花費在ISA總線上。大多數總線控制的Linux驅動程序現在使用一種“copybreak”方案，較大的數據包直接從網卡放進內核的網絡緩存，小的數據包被CPU復制到cache里進行下一步的處理。

缺點：（只適用于ISA總線的網卡）網卡要求低端內存緩存和一個DMA通道。任何總線控制器在與其它強占總線的總線控制器，如某些古老的SCSI適配器，一起工作時都會出問題。有几個設計低劣的主板芯片組在與總線控制器一起使用時也有麻煩。不使用任何類型的DMA設備的一個原因是使用了為代

替386設計的486處理器插件：這些處理器在每個DMA周期都必須刷新cache。（這其中包括Cx486DLC、Ti486DLC、Cx486SLC、Ti486SLC，等等。）

7.2 涉及總線帶寬的性能

ISA總線能達到5.3MB/sec (42Mb/sec)，聽起來對10Mbps以太網是足夠了。對于100Mbps網卡，顯然需要更快的總線以利用網絡帶寬。

7.2.1 8比特ISA網卡與16比特ISA網卡

你可能不會再買一個新的8比特ISA以太網卡了，但在隨后几年間你會在計算機物物交換市場一類的地方發現很多很便宜的這種卡。這使得它們在“家用以太網”系統中很常見。對于16比特的ISA網卡來說也是一樣的，因為現在PCI卡已經很普遍了。

對于較低或平均的應用，一些可以提供足夠性能的8比特網卡是wd8003、3c503和ne1000。3c501的性能較差，而且要避免使用那些12年前XT時代的低性能紀念品。（可以把它們送給Alan，他在收集這種網卡......）

8比特的數據通道不會對性能造成太大損害，你依然可以從一個較快主機上的8比特wd8003網卡（使用快的ISA總線）上獲得大約500到800kB/s的ftp下載速度。如果你的所有網絡流量几乎都是來自遠端站點，那么數據通道上的瓶頸將發生在其它地方，只有在進行本地子網操作時你才會注意到不同網卡間的速度差別。

7.3 32比特(VLB/EISA/PCI)以太網卡

注意，10Mbs網絡通常并不一定要求32比特的接口。參見7.1 (可編程I/O、共享內存與DMA)以了解為什么8MHz的ISA總線上的10Mbps以太網卡并非真正的瓶頸。盡管使用快的總線的以太網卡并不一定帶來更快的數據傳輸，但會減少CPU負載，這對多用戶系統來說是件好事。當然，對于現在普通的100Mbps網絡，32比特接口是充分使用全部帶寬的必要條件。

7.4 編寫驅動程序

在Linux下使用以太網卡所必需的只不過是相應的驅動程序。因此，關鍵是制造商要向公眾公開編程的技術資料，而無需你（或其他什么人）簽署什么協議。關于獲取資料的可能性（也許你不編寫代碼，那么就是其他人編寫你確實需要的驅動程序的可能性），一個較好的指南是Crynwr (昵稱Clarkson)的包驅動程序的可用性。Russ Nelson在干這些事，對開發Linux驅動程序很有幫助。網上沖浪者可以試著用下面的URL看一下Russ的軟件。

Russ Nelson's Packet Drivers `<http://www.crynwr.com>`

有了資料，就可以為網卡編寫驅動程序并在Linux下使用（至少從理論上來說是這樣）。記住，有些為XT一類機器設計的老式硬件在Linux這樣的多任務環境下工作得不是很好。如果網絡流量較大，使用這些網卡會帶來大麻煩。

大多數網卡都帶有如NDIS和ODI一類的MS-DOS接口的驅動程序，但對Linux沒有用。許多人建議直接鏈接它們或自動翻譯一下，但這几乎是不可能的。MS-DOS驅動程序需要工作在16比特模式，并依賴于“軟件中斷”，這二者與Linux內核不兼容。這種不兼容實際上是Linux的一個特性，有些Linux驅動

程序比其相應的MS-DOS驅動程序要好得多。比如“8390”系列驅動程序使用乒乓傳送緩存，該方法剛剛被引進MS-DOS。

（乒乓傳送緩存意味著為傳送數據包使用至少兩個最大大小的包緩存。在網卡發送其中的一個時，載入另一個。在第一個包被發出去后，立刻發送第二個包，依次類推。這樣，大多數網卡就可以連續向線路上發送背靠背數據包。）

好啦。你可以決定為Foobar Ethernet網卡編寫驅動程序了，因為你有編程資料，而且還沒人寫這個驅動程序。（......這是兩個主要的需求 ;-) 你可以從Linux內核源碼樹中提供的網絡驅動程序框架開始。在所有近期的內核里都能找到這個文件/usr/src/linux/drivers/net/skeleton.c。在2.4.x（或更新的）內核里已經把它重新命名為`isa-skeleton.c`。也可以看看如下URL的Kernel Hackers Guide： *KHG* `<http://www.redhat.com:8080/HyperNews/get/khg.html>`

7.5 內核的驅動程序接口

下面對編寫一個新驅動程序所必需的函數進行了若干說明。和上面提到的驅動程序框架一起閱讀可以更清楚一些。

7.5.1 Probe

在啓動時調用以檢查網卡存在與否。如果可以通過讀取內存等非强制手段進行檢查最好。也可以從I/O端口讀取。在探測開始向I/O端口寫不好，因為這樣可能會損害另一個設備。通常在這里還進行一些設備初始化（分配I/O空間、IRQ、填充dev->???域等等）。必須了解網卡可以配置到哪些I/O端口/內存、如何啓用共享內存（如果用了的話）以及如何選擇/啓用中斷產生，等等。

7.5.2 Interrupt handler

在網卡發出一個中斷時內核調用的程序。它需要確定網卡發出中斷的原因并進行相應的操作。一般的中斷條件是接收到數據、發送完成、報告出錯狀況。需要了解相關的中斷狀態位以進行相應的操作。

7.5.3 Transmit function

與dev->hard_start_xmit()鏈接，在內核想通過設備傳送數據時調用它。該函數把數據放入網卡并觸發傳送。需要了解如何把數據打包并傳給網卡（共享內存拷貝、PIO傳送、DMA？），以及放入網卡正確的位置。然后需要了解如何通知網卡把數據發送到線路上，（可能）在發送完成后發出一個中斷。在硬件無法接收更多數據包時需要設置dev->tbusy標志。在網卡有空間可用時，一般這發生在傳送完成中斷過程中，清除dev->tbusy標志并用`mark_bh(INET_BH)`通知上一層。

7.5.4 Receive function

在網卡報告有數據時由內核中斷處理程序調用。它把數據從網卡上移出，放入一個sk_buff并通過執行netif_rx(sk_buff)告訴內核數據所在位置。需要了解如何在接收數據時啓用中斷生成，如何檢查相關的接收狀態位，以及如何從網卡獲取數據（通過共享內存拷貝、PIO、DMA，等等）。

7.5.5 Open function

與dev->open鏈接，在有人使用`ifconfig eth0 up`時網絡層調用它——把設備連到線路上并啓用來接收/發送數據。任何在探測過程中（啓用IRQ生成函數等）沒有完成的特別的初始化操作都在這里進行。

7.5.6 Close function （可選）

在有人使用`ifconfig eth0 down`時使網卡進入一個清醒的狀態。如果硬件許可的話它會釋放中斷和DMA通道，并完全關閉以節約能源（象收發器一樣）。

7.5.7 Miscellaneous functions

如同一個重新設置函數，如果事情變得很糟，驅動程序可以試圖重新設置網卡作為最后防線。一般在發送超時或類似情況下如此進行。也是一個讀取網卡統計寄存器的函數，如果是這樣配備的話。

7.6 3Com的技術信息

如果對3Com網卡驅動程序的工作感興趣，可以從3Com公司獲取技術資料。Cameron在下面友好地告訴了我們該如何去做：

在我們的“技術參考文獻”（TRs）里給出了3Com的以太網適配器驅動程序程序員需要了解的資料。這些手冊描述了板上的程序員接口，但沒有提及診斷、安裝程序等終端用戶所看到的東西。

網絡適配器分部的市場部有技術參考資料分發。為了使這個計划更有效，我們把它集中到一個稱作“CardFacts”的自動電話系統里。你可以打電話來，然后它把資料傳真給你。要索取技術參考資料，打電話到408-727-7021。索取開發人員的訂單，資料號是9070。在打電話前准備好你的傳真號碼。填完訂單后把它傳真到408-764-5004。手冊會由聯邦速遞的次日服務送到。

有人認為我們的手冊不該免費，他們也在尋找此系統過于昂貴或占用的時間和努力太多的証據。到目前為止，3Com的顧客確實不錯，向我們提出的要求也很合理。我們需要你們的繼續合作并把這樣的服務維持下去。

7.7 基于AMD PCnet/LANCE的網卡的注意事項

AMD的LANCE（以太網的局域網控制器）是最早提供的，已經被“PCnet-ISA”芯片所取代，否則又名79C960。注意，“LANCE”名稱有毛病，有些人會用老名稱稱呼新芯片。AMD的網絡產品分部的Dave Roberts友好地提供了下面有關該芯片的信息：

“從功能上來看，它等同與NE1500。它的寄存器組與使用附加1500/2100結構的老式LANCE一樣。PCnet-ISA可以使用較早的1500/2100驅動程序。NE1500和NE2100的結構基本上是相同的。開始Novell把它稱為2100，但后來想區分同軸電纜與10BASE-T網卡。屬于10BASE-T的就只采用1500范圍的編號。這是僅有的區別。

許多公司提供基于PCnet-ISA的產品，包括HP、Racal-Datacom、Allied Telesis、Boca Research、Kingston Technology等等。除了有些制造商增加了“無跳線”特性允許軟件配置網卡外，這些網卡基本上都是一樣的。大多數制造商沒有增加這一特性。AMD提供了一個使用PCnet-ISA的網卡的標准設計軟件包，許多制造商不加改變地直接使用我們的設計。這也就是說，如果想編寫大多數基于PCnet-ISA的網卡的驅動程序，只需要從AMD獲取數據資料。打電話給我們的資料分發中心(800)222-9323，索取PCnet-ISA的數據資料Am79C960。這是免費的。

要迅速了解一塊網卡是否“標准”網卡只需要看一下。如果是標准的，網卡上只有一塊大的芯片、一塊晶振、一塊小的IEEE地址PROM、可能還有一個啓動ROM的插座和一個連接器（依照提供的媒介選項可能是1、2或3）。注意，如果是同軸電纜網卡，卡上就應該有一些收發器緩存，它們靠近連接器，遠離PCnet-ISA。”

一個可能的網卡黑客需要注意，不同的LANCE產品采用不同的“重起”方法。有些恢復到上次離開網絡環路的地方，另一些則從環路的開頭開始，就象剛被初始化一樣。

7.8 廣播與混雜模式

Donald所做的另一個工作是實現廣播與混雜模式的鉤子函數。所有發布的（即不是ALPHA版本的）ISA驅動程序現在都支持混雜模式。

Donald寫道：“我准備從討論混雜模式開始，它從概念上來説很容易實現。對大多數硬件，你只需要設置一個寄存器位，然后就可以接收到線路上的每一個數據包。對，差不多就這么簡單；對有些硬件，你必須先關閉板卡（可能會丟失若干數據包），重新配置它，然后重新啓用以太網卡。對吧，就這么簡單，下面要討論的就不是這么明顯了：廣播模式。它可以用兩種方式實現：

1. 使用混雜模式和一個如Berkeley包過濾器（BPF）的數據包過濾器。BPF是一個模式匹配指令語言，可以編寫一個程序挑出感興趣的地址。它的優點在于它很普遍和可編程。其缺點是沒有一個一般性的方法可以讓内核避免打開混雜模式和通過每一個注冊的包過濾器運行每一個線路上的數據包。參見7.9 (Berkeley包過濾器)以了解更多信息。
2. 使用絶大多數以太網芯片内建的廣播包過濾器。

我想應該列出几個以太網卡/芯片提供的廣播包過濾器：

```
芯片/網卡          混雜模式     廣播包過濾器
-------------   ------------    ---------------
Seeq8001/3c501  Yes                         Binary filter (1)
3Com/3c509                Yes                         Binary filter (1)
8390                      Yes                         Autodin II six bit hash (2) (3)
LANCE                     Yes                         Autodin II six bit hash (2) (3)
i82586                    Yes                         Hidden Autodin II six bit hash (2) (4)
```

1. 這些網卡聲稱有一個過濾器，但只是簡單地對“accept all multicast packets”或“accept no multicast packets”回答yes/no。
2. AUTODIN II是標准的以太網CRC校驗多項式。在這種方式下，廣播地址被哈希運算后在哈希表里進行查找。如果啓用了相應的比特位，則數據包被接收。以太網數據包的設計使得硬件在如此處理時的開銷很小——（通常）只要在前6個八進制數（目標地址）之后鎖定CRC電路（用來進行錯誤檢查）的6個比特位，把它們作為哈希表的索引（6比特——一個64比特的表）。
3. 這些芯片使用6比特哈希，必須由主機計算并載入哈希表。這也就是説内核必須包含CRC代碼。
4. 82586内部使用6比特哈希，但是由自己從接受的廣播地址列表計算出哈希表。

注意，這些芯片的過濾效果都不好，還需要一個中間層次的模塊完成最后的過濾。同時還要注意，在每種情況下都必須保持一個完整的接受廣播地址列表，在出現變化時以重新計算哈希表。

7.9 Berkeley包過濾器（BPF）

開發者普遍認為BPF的功能不該由內核提供，而是放在一個（希望很少用到的）兼容庫里。

不了解的人來說：BPF（Berkeley包過濾器）是一種向內核網絡層説明對哪些數據包感興趣的機制。它是用一種建立在底層網絡代碼中的特殊指令語言解釋器實現的。應用程序把一個用這種語言編寫的程序傳遞給內核，然后內核對每一個接收到的數據包執行該程序。如果內核有多個BPF應用程序，對每個數據包都要運行這個程序。

問題在于很難從數據包過濾器程序推斷出應用程序實際上對哪一種數據包感興趣，所以一般的解決方法就是始終運行過濾器。假設一個應用程序注冊的BPF程序是獲取發往某個廣播地址的低速數據流。絕大多數以太網卡有一個64個入口的哈希表的硬件實現的廣播地址過濾器，用來忽略大多數不想要的廣播數據包，所以有可能以極低的開銷完成這一操作。但是由于有了BPF，內核必須把接口設置為混雜模式，接收所有數據包，并對它們運行過濾器。不管怎樣，這樣確實可以工作，但考慮到對所要求的數據包進行的處理，就已經變得過于麻煩了。

8 便攜式/筆記本電腦聯網

有几種方法把筆記本電腦聯上網。可以使用SLIP代碼（并以串行線速率運行）；可以使用一個有內建支持PCMCIA插槽的筆記本電腦；可以使用一個有插座并插入了一塊ISA網卡的筆記本電腦；或者使用并口以太網適配器。

8.1 使用SLIP

這是最便宜的解決方案，但最麻煩。另外，無法獲得很高的傳輸速率。因為SLIP實際上與以太網卡無關，這里就不多説了。參見NET-2 Howto。

8.2 PCMCIA支持

設法完全確定硬件的情況（即網卡制造商、PCMCIA芯片控制器的制造商），然后在筆記本電腦討論組里詢問。不管怎樣，不要把事情想得這么簡單。預料會花費一些時間、給內核打補丁等。可能有一天只需要鍵入“make config”。8-)

目前所支持的兩個PCMCIA芯片組是Databook的TCIC/2和Intel的i82365。

在tsx-11.mit.edu的/pub/linux/packages/laptops/下有一些程序可能會有用。它們包括從PCMCIA以太網卡驅動程序到與PCMCIA控制器芯片進行通信的程序。注意，這些程序一般與某個特定的PCMCIA芯片（即intel 82365或TCIC/2）相關。

對于NE2000兼容卡，有人通過在DOS下配置網卡，然后在DOS命令提示符下執行`loadlin`啓動Linux，成功地實現了對PCMCIA的支持。

目前一直在獲得實質性的進展，對于希望PCMCIA支持的Linux用戶，情況正在好轉。這一努力的先驅是David Hinds。他的最新PCMCIA支持軟件包可以從下面得到：

PCMCIA Package `<ftp://cb-iris.stanford.edu/pub/pcmcia>`

尋找一個類似`pcmcia-cs-X.Y.Z.tgz`的文件，其中X.Y.Z是最近的版本號。很有可能它也被上載到了FTP站點`tsx-11.mit.edu`。

注意，Donald的PCMCIA啓用程序是一個用戶級的程序，而David Hinds的程序是運行在內核級。最好使用David的軟件包，因為它的使用很廣泛，并在持續開發過程中。

8.3 對接座內的ISA以太網卡。

筆記本電腦的對接座一般要花$250，提供兩個全尺寸的ISA插槽、兩個串口和一個并口。大多數對接座使用筆記本電腦的電池供電，如果使用短的ISA卡，有若干對接座可以使用對接座里附加的額外電池供電。你可以加一個便宜的以太網卡，并享受全速的以太網性能。

8.4 袖珍/并口適配器。

“袖珍”以太網適配器也可以滿足需要。注意，由于并口的接口限制，傳輸速率不夠好（可能最高是200kB/s？）。

還有一個約束是牆磚一樣的電源。你可以購買/制作一個從筆記本電腦的鍵盤端口引出電源的電纜，擺脫與適配器連在一起的牆磚。（參見4.31.1 (鍵盤電源)）

參見4.14.5 (DE-600 / DE-620)和4.31.1 (RealTek)以了解支持的兩款袖珍適配器。

9 雜項

所有不適合放在別處的相關信息都堆在這里。可能不相干，也可能大家不感興趣，但還是放在這兒了。

9.1 向內核傳遞以太網參數

有兩個通用的內核命令可以在啓動時向內核傳遞以太網參數（`ether` 和`reserve`）。可以用LILO、loadlin或其它接受可選參數的啓動工具完成該操作。

例如，如果命令為“blah”，希望接收3個參數（假定為123、456和789），那么在使用LILO時就應該如下：

`LILO: linux blah=123,456,789`

要了解啓動時參數的更多信息（和完全的列表），請參見 *BootPrompt-HOWTO* `<http://metalab.unc.edu/mdw/HOWTO/BootPrompt-HOWTO.html>`

9.1.1 ether命令

`ether=`參數與直接構建在內核的驅動程序一起使用。`ether=`參數對一個模塊化的驅動程序完全不起作用。它的最通用形式如下：

```
ether=IRQ,BASE_ADDR,PARAM_1,PARAM_2,NAME
```

所有的參數都是可選的。第一個非數字的參數被用做NAME。

IRQ: 很明顯。為“0”的IRQ值（一般為缺省值）意味著autoIRQ。首先設置IRQ而不是base_addr是一個歷史性的巧合——無論在改變別的什么時都可以順便改正這一點。

BASE_ADDR: 也很明顯。值為“0”（一般為缺省值）意味著探測以太網卡的網卡類型特定的地址列表。

PARAM_1: 這開始是用來覆蓋WD80*3這樣的共享內存網卡的起始內存的值。有些驅動程序使用該值的低4位來設置診斷信息級別。0——缺省值，1-7——級別1..7，（7是最完全的信息）8——級別0（沒有信息）。另外，LANCE驅動程序使用該值的低4位來選擇DMA通道。否則就使用auto-DMA。

PARAM_2: 3c503驅動程序使用它來選擇內部還是外部收發器。0——缺省/內部，1——外部AUI。Cabletron的E21XX網卡還使用PARAM_2的低4位來選擇輸出媒介。否則就自動檢測。

NAME: 選擇該值所指的網絡設備。標准內核對附屬于總線的以太網卡使用名稱“eth0”、“eth1”、“eth2”和“eth3”，對并口“袖珍”以太網適配器使用“atp0”。arcnet驅動程序使用名稱“arc0”。可以使用這些LILO參數明確設置基址來啓用多塊網卡。1.0內核把基于LANCE的以太網卡作為特殊情況進行處理：LILO參數被忽略，LANCE網卡總是被分配為從“eth0”開始的名稱“eth<n>”。附加的非LANCE網卡必須被明確指定為“eth<n+1>”，并用諸如“ether=0,-1,eth0”的方式禁止通常的“eth0”探測。（對，這是個Bug。）

9.1.2 reserve命令

這個緊接著的LILO命令用法與上面的“ether=”一樣，即附加在lilo.conf里指定的啓動選擇名稱后面。

```
reserve=IO-base,extent{,IO-base,extent...}
```

在某些機器上，可能需要防止設備驅動程序在某個特定區域里檢查設備（自動探測）。其原因可能是由于設計低劣的硬件使啓動凍結（如某些以太網卡）、被錯誤識別的硬件、在較早的探測中狀態被改變的硬件、或者僅僅是不想讓內核初始化硬件。

啓動時的參數`reserve`通過指定無需探測的I/O端口區域來解決這個問題。該區域保留在內核的端口注冊表里，就象該區域里已經發現了一個設備一樣。注意，這一機制在大多數機器上是不必要的。只有在有問題或特定的情況下才有必要使用它。

指定區域里的I/O端口受到保護，不被設備探測影響。在某些驅動程序被NE2000挂起，或其它設備被錯誤地識別為NE2000時使用這一方法。正確的設備驅動程序不該探測一個保留區域，除非另一個啓動參數明確指定它這么做。這隱含了`reserve`經常與其它啓動參數一起使用。因此，如果指定了一個`reserve`區域來保護某個特定設備，通常就必須明確指定對該設備的探測。大多數驅動程序在給定了明確地址后就忽略了端口注冊表。

例如，啓動行

```
LILO: linux reserve=0x300,32 ether=0,0x300,eth0
```

使以太網卡驅動程序以外的所有設備驅動程序都不探測0x300-0x31f。

一般啓動時的參數限制是11個，因此每個`reserve`關鍵詞只能指定5個保留區域。如果請求很復雜，可以使用多個`reserve`指令。

9.2 把以太網驅動程序作為模塊使用

現在大多數Linux發行版里的內核都只包含很少几個內建的驅動程序。驅動程序都采用獨立的動態可加載模塊的形式提供。這些模塊化驅動程序通常由管理員使用`modprobe(8)`命令載入，或者在某些情況下由內核通過“kerneld”（在2.0版）或“kmod”（在2.1版）自動載入，然后調用`modprobe`。

你所用的發行版可能會提供良好的圖形配置工具來設置以太網模塊。如果可能就先使用它們。下面描述了在這些花俏的配置程序下的信息，以及這些程序改變了哪些東西。

控制使用哪些模塊和每個模塊提供哪些選項的信息一般保存在文件/etc/conf.modules里。在這個文件里使用的兩個感興趣的主要選項（對以太網卡而言）是alias和options。modprobe命令查閱該文件以了解模塊信息。

實際的模塊本身一般保存在名為/lib/modules/‘uname -r‘/net的目錄下，其中uname -r命令給出內核的版本（如2.0.34）。你可以在這里看看哪一個模塊與你的網卡匹配。

在你的conf.modules文件里首先需要的是告訴modprobe對于eth0（和eth1以及......）網絡接口使用什么驅動程序。為此要使用

```
alias
```

命令。例如，有一塊使用smc-ultra.o驅動程序模塊的ISA SMC EtherEZ網卡，需要增加如下一行把該驅動程序alias到eth0上：

```
        alias eth0 smc-ultra
```

其次，可能需要一個options行來指出與某個特定模塊（或模塊別名）一起使用哪些選項。還采用上面那個例子，如果只有一行alias而沒有options行，內核會警告（參見dmesg）說自動探測ISA網卡不好。要消除這個警告，需要增加另一行來告訴模塊網卡被配置在哪個I/O基址上，比方說是16進制地址0x280。

```
        options smc-ultra io=0x280
```

大多數ISA模塊在insmod命令行接受io=0x340和irq=12這樣的參數。提供這些參數以避免探測該網卡是必須的或至少是强烈建議的。與PCI和EISA設備不同，對大多數ISA設備而言沒有真正安全的自動探測方法，所以在把驅動程序作為模塊使用時應當避免自動探測。

每個模塊所接受的所有選項列表可以在以下文件中找到：

/usr/src/linux/Documentation/networking/net-modules.txt

推荐閱讀該文件以了解對你的特定網卡可以使用哪些選項。注意，對于單個模塊能夠處理多個設備的模塊，有些支持用逗號分開的值列表，如所有基于8390的驅動程序和PLIP驅動程序。例如：

```
        options 3c503 io=0x280,0x300,0x330,0x350 xcvr=0,1,0,1
```

上面就是一個模塊控制四塊3c503網卡，其中網卡2和4使用外部收發器。不要在“=”或逗號周圍使用空格。

還要注意，一個忙的模塊不能被刪除。這也就是說在刪除模塊前需要使用ifconfig eth0 down（關閉以太網卡）。

命令lsmod可以顯示哪些模塊被載入、它們是否正在被使用，而rmmod可以刪除這些模塊。

9.3 相關文檔

本文檔的很多信息來自comp.os.linux新聞組保存的帖子，這表明該新聞組是一個有價值的信息源。其它的有用信息來自Donald自己的一大堆小文件。當然，如果你正在設置一塊以太網卡，你還要讀一下NET-2 Howto才能真正配置好你要用的軟件。另外，如果你想做個黑客，你總能從驅動程序的源碼里找到些有用的信息。在開始實際代碼之前，總有一到兩段描述關鍵點的文字。

對于那些尋找不是Linux特有信息（如什么是10BaseT、什么是AUI、Hub是干什么的等等）的人，我强烈推荐使用新聞組*comp.dcom.lans.ethernet*或*comp.sys.ibm.pc.hardware.networking*。那些在`dejanews.com`上的新聞組檔案也是很有價值的信息源。你可以從以下URL的RTFM（保存所有新聞組的FAQ）獲取新聞組的FAQ：

Usenet FAQs `<ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/>`

你還可以在以下URL看一下可以說是“Ethernet-HomePage”的主頁：

Ethernet-HomePage `<http://wwwhost.ots.utexas.edu/ethernet/ethernet-home.html>`

9.4 聲明與版權

本文檔不是福音書。它可能是你所找到的最新的資料。除了你自己，沒人會對你的硬件出現的問題負責。如果你的以太網卡或其它的硬件冒煙的話（几乎不可能！），我們也不承擔任何責任。也就是說，本文作者不對任何使用本文信息采取行動造成的損失負責。

對翻譯者的提示。首先，翻譯SGML源文件（可以用FTP從HowTo主站點獲得），這樣可以生成其它的輸出格式。請注意保留一份翻譯的原始英文SGML源文件！在發布更新后的HowTo時，獲取該版本的新SGML源文件，簡單地使用`diff -u old.sgml new.sgml`就可以知道何處有變動，并把這些變動放入你的翻譯版本，無需重新閱讀或重新翻譯全部內容。

如果你想在公開出版物中使用本文檔，請聯系本文作者（通過e-mail），這樣可以獲得最新的資料。以前有些Linux HowTo的過時版本被公開出版，使得開發者為一些在更新版本中已經回答過的問題飽受折磨。

This document is *not* gospel. However, it is probably the most up to date info that you will be able to find. Nobody is responsible for what happens to your hardware but yourself. If your ethercard or any other hardware goes up in smoke (...nearly impossible!) we take no responsibility. ie. THE AUTHORS ARE NOT RESPONSIBLE FOR ANY DAMAGES INCURRED DUE TO ACTIONS TAKEN BASED ON THE INFORMATION INCLUDED IN THIS DOCUMENT.

A hint to people considering doing a translation. First, translate the SGML source (available via FTP from the HowTo main site) so that you can then generate other output formats. Be sure to keep a copy of the original English SGML source that you translated from! When an updated HowTo is released, get the new SGML source for that version, and then a simple `diff -u old.sgml new.sgml` will show you exactly what has changed so that you can easily incorporate those changes into your translated SMGL source without having to re-read or re-translate everything.

If you are intending to incorporate this document into a published work, please make contact (via e-mail) so that you can be supplied with the most up to date information available. In the past, out of date versions of the Linux HowTo documents have been published, which caused the developers undue grief from being plagued with questions that were already answered in the up to date versions.

9.5 結束語

如果你在本文檔中發現有錯別字或過時信息，請給我發e-mail。本文檔比較大，所以很容易出現疏漏。如果你曾發e-mail要求改正，而在新版本中依然沒改過來，請把改正要求再發一遍，可能你的上一封e-mail被我平時收到的大量垃圾信件掩蓋了。

謝謝！

Paul Gortmaker, `p_gortmaker@yahoo.com`（戈紹男，`isaac_ge@yahoo.com`）