The Linux Printing HOWTO 中文版

作者：Grant Taylor <*gtaylor+pht@picante.com*>
譯者：任恆宇 <*henryjen@mail.net.tw*>

v3.29, 9 March 1999 翻譯日期：15 六月 1999

這一篇是 Linux Printing HOWTO 的中譯本, 內容包括了如何在 Linux 之下產生, 預覽,列印及傳真任何文件的相關資訊 (基本上亦可適用於其他的 Unices).

目錄

1 簡介

這份 Printing HOWTO 應該包括了能幫助你在 Linux 平台上架設列印服務的所有訊息. 這不像在 Microsoft 或 Apple 環境下用滑鼠點選那麼容易, 但一定比管理一個大型的區域網路來得輕鬆.

這份文件的編排方式是希望能讓大部份的使用者只須閱讀前半段. 較難理解的個案情況則擺在後半段, 而且你可以很容易地從目錄中找到. 所以大部份人可能只須看到第九或十節就可以了.

因爲 3.x 版的文件重新寫過, 前面版本中的許多內容已經拿掉了. 這是故意的, 因爲前版的 HOWTOs 長達 60 多頁, 而且編排方式並不是很理想. 如果你在這份文件中找不到所需的答案, 你可以 a) 參閱位於 *Printing HOWTO Home Page* `<http://www/picante.com/~gtaylor/pht/>` 的舊版文件. 也可以 b) 告訴我缺少了什麼部份.

你可以在 *Printing HOWTO Home Page* `<http://www.picante.com/~gtaylor/pht/>` 找到這份文件的最新版本; 當然你也可以在 Metalab (`metalab.unc.edu`) 和附近的 LDP mirror 站找到.

1.1 歷史

這是第三代的 Printing HOWTO, 也就是第三次的完全改版, 以年份來看:

1. 我在 1992 年因爲看到在 comp.os.linux 有很多關於列印的問題, 便寫下並發表了 printing-howto. 比 HOWTO 計畫還早幾個月, 同時也是第一份稱爲 'howto' 的常見問答集. 這一版是純文字格式.

2. 加入了 HOWTO 計畫後, Printing-HOWTO 和由 Brian McCauley `<B.A.McCauley@bham.ac.uk>` 所寫的 Lpd FAQ 合併; 我們繼續合寫 PHT 差不多兩年. 有些地方則納入了 Karl Auer `<Karl.Auer@anu.edu.au>` 的作品. 這一版的 PHT 是以 TeXinfo 寫成, 也提供了 PS, HTML, Ascii 和 Info 的版本.

3. 在 PHT 經過一年多的時間逐漸成熟以及一次嘗試移轉給別人維護的失敗經驗後, 產生了這份重新改版. 這一版的 PHT 是用 LinuxDoc DTD 和 SGML-Tools-1 套件以 SGML 所寫成的. 從 3.27 版開始, 納入了一份支援印表機資料庫的摘要; 在 3.27 版之前, 這份 HOWTO 中從來沒有一份相容印表機清單 (!).

1.2 版權聲明

2 如何列印

如果你或你的系統管理者或廠商已經設定好 lpd 的話, 你所需要做的便只是學習如何使用 lpr 指令而已. *Printing Usage HOWTO* `<http://metalab.unc.edu/LDP/HOWTO/Printing-Usage-HOWTO.html>` 包含了這部份的資訊, 還有一些你可能應該要知道的列印佇列處理指令.

如果你有一個全新系統或者新的印表機, 那你就必須在開始列印前先設定好列印服務. 如果是這樣, 請繼續看下去!

3 核心印表機裝置

平行埠有兩種完全不同的裝置驅動程式; 你的核心版本決定了你用的是那一種. 驅動程式是在 Linux 2.1.33 版所做的改變.

兩種驅動程式的型態僅在很少的一些細節上相同. 最值得注意的是, 許多人發現 Linux 不會偵測他們的平行埠除非他們在 PC BIOS 的設定中把 "Plug and Play" 關掉. (這一點也不令人驚訝; 過去的記錄顯示在 Windows 和其它地方對非 PCI 裝置的 PnP 已經成了某種災難).

3.1 lp 裝置 (核心版本 <=2.1.32)

假如你已經編譯或載入了 lp 裝置(如果載入了的話, `cat /proc/devices` 的結果應該會看到 lp), Linux 核心 (<=2.1.32) 會提供一個或多個的 */dev/lp0*, */dev/lp1*, */dev/lp2* 裝置檔案. 這些檔案不是隨意指定的, 相反的, 每一個檔案皆對應到一個特定的 I/O 位址. 這表示你的第一部印表機可能是 *lp0* 或 *lp1*, 完全視你的硬體情況而定. 兩個都試試看吧!

有些使用者提到如果使用較舊的單向印表機連接線, 就無法偵測到他們的雙向印表機連接埠. 確定你使用了正確的連接線.

一個印表機埠不能同時載入 plip 和 lp 的驅動程式, 如果你用的是 2.x 或是 1.3.x 版以後的核心, 你可以用手動或核心的方式選擇載入不同的驅動程式. 只要小心的設定中斷等設定, 你應該可以在一個埠上執行 plip, 而在另一個埠上載入 lp 的驅動程式. 曾經有人修改驅動程式而成功; 我希望能早日聽到有人以一個簡單的指令完成相同的事情.

有一個小程式 *tunelp*, 如果你是 root, 可以用來設定 Linux 2.0 lp 裝置使用的中斷 (interrupt), 輪詢速率 (polling rate) 和其它的項目.

若是核心編譯時包括了 lp 驅動程式, 核心接受 `lp=` 參數設定中斷及 IO 位址:

```
若將 lp 驅動程式編譯入核心, 你可以用 LILO/LOADLIN 的命令列來指定使用的
IO 位址及中斷.
```

語法： lp=port0[,irq0[,port1[,irq1[,port2[,irq2]]]]]

範例： lp=0x378,0 或 lp=0x278,5,0x378,7 **

要注意的是假如你使用這個功能，你必須指定 *所有* 要用的印表機埠，
系統不提供預設值．你可以用 lp=0 來停用內建的驅動程式．

若是以載入模組的方式執行, 可以在 insmod 指令的命令列(或是在 */etc/conf.modules* 來模擬核心模式) 以平常模組參數的語法來指定 IO 位址和中斷. 參數爲 io=port0,port1,port2 和 irq=irq0,irq1,irq2. 請參考 insmod 的線上手册以取得更進一步的資訊.

**給那些當需要時卻總是找不到標準印表機埠位址的人, 位址便如同第二個範例中所示, 另一個印表機埠 (*lp0*) 則在 0x3bc. 我不知道一般所用的中斷是那一個.

Linux 2.0 並列埠驅動程式的原始碼在 /usr/src/linux/drivers/char/lp.c.

3.2 parport 裝置 (核心版本 >= 2.1.33)

從 2.1.33 版本後的核心(2.0.30 版核心可取得修正檔), lp 裝置成爲新增加的 parport 裝置的客户端. 增加 parport 裝置解決了許多困擾舊有 lp 裝置驅動程式的問題 - 像是可以和其它的驅動程式共用一個連接埠, 可以動態指定可用的並列埠而不須將裝置檔對應到固定的 I/O 位址等等.

parport 裝置的出現使得很多新的並列埠設備, 像是 Zip 磁碟機, 可攜式 CD-ROM 和磁碟機等等, 的驅動程式得以出現. 其中一些也有提供 2.0 核心可用的版本, 到網路上找找看吧.

你會注意到的主要差異, 以目前列印來看, 是基於 parport 的核心會動態地指定 lp 裝置給平行埠. 所以在 Linux 2.0 中的 lp1 在 Linux 2.2 中可能是 lp0. 如果你從使用 lp 驅動程式的核心升級到使用 parport 驅動程式的核心, 記得檢查這一點.

當我使用了 parport 驅動程式後, 我會更完整地探討它. 目前你可以看一下核心原始檔中的 *Documentation/parport.txt*. 或者你也可以到 *parport web site* <http://www.cyberelk.demon.co.uk/parport.html> 看看.

3.3 序列裝置(Serial devices)

序列裝置在 Linux 中通常有個像 */dev/ttyS1* 這類的名稱. *stty* 這個工具程式可以讓你查看及設定序列埠. *setserial* 可以讓你調整更多的屬性及設定使用非標準的 IRQ 及 I/O 位址的序列埠. *Serial-HOWTO* <http://metalab.unc.edu/mdw/HOWTO/Serial-HOWTO.html> 會對 Linux 下的序列埠做更進一步的探討.

如果你使用的是有流量控制的慢速序列印表機, 可能會發現有些報表被截斷. 這是因爲系統預定在程式關閉序列埠 30 秒後, 所有在緩衝區內尚未傳送的字元都會被清除. 緩衝區內可容納最多 4096 個字元, 假使你的印表機使用流量控制, 而且慢到無法在程式關閉序列埠後 30 秒內印出緩衝區內的所有字元, 那麼緩衝區內最後的內容就會遺失. 如果 cat file > /dev/ttyS2 這個指令可以完整印出較小的檔案, 但會截斷較長的檔案時, 你可能就是遇到了這種情況.

30 秒的週期可以用 setserial (2.12 及以後的版本) 指令的 "closing_wait" 選項來調整. 一台機器通常是在 rc.serial 開機檔中呼叫 setserial 來初始化序列埠. 可以在這修改連接印表機的序列埠的 closing_wait 選項.

4 支援的印表機

Linux 核心大致上支援所有能夠接到序列或並列埠的印表機, 但還是有些要注意的事項, 和一些雖然能與 Linux 溝通(就電子訊號而言) 卻無法使用的印表機. 與 Linux 不相容的主要是那些被稱爲 "Windows" 或是 "GDI" 印表機的機種. 它們之所以會被這樣稱呼是因爲部份, 甚至全部的印表機控制語言和列印機制的設計細節都缺乏說明文件. 通常這種廠商會提供 Windows 的驅動程式, 而且也很滿足於只賣給 Windows 的使用者; 這就是爲什麼它們被稱爲 Winprinters. 某些時候廠商也會提供 NT, OS/2 或其它作業系統的驅動程式.

許多這種印表機在 Linux 下不能用. 有少部份的可以, 也有一些只有部份功能可用 (通常是由於有人針對寫驅動程式所需的細節進行反向工程). 請參閱後面的印表機支援清單取得特定機種的資訊.

有少數的印表機是介於中間的. 舉例來說, 有些 NEC 的機種使用標準印表機語言 PCL 中較簡單的功能, 讓懂 PCL 的軟體可以達到列印 300 dpi 的水準, 而只有 NEC 才知道如何讓這些印表機發揮 600 dpi 的列印能力.

萬一你已經有這種印表機, 仍然有變通之道, 只不過相當的笨拙, 所以我絕對不會使用這種方式. 想更深入的瞭解 Windows-only 印表機請看第 12 節

4.1 Postscript

那些印表機一定可以用呢? 最好的選擇就是買具備 PostScript 支援的印表機. 幾乎所有的 Unix 軟體都以 PostScript 來產生報表, 所以很明顯的, 如果能有一部直接支援 PostScript 的印表機的話是再好不過了! 不幸的是除了雷射印表機外, 其它種類的印表機幾乎都不支援 PostScript, 要不然就是相當昂貴的選購配備.

Unix 下的軟體和一般印刷業之所以選擇 PostScript 做爲印表機控制語言的標準是有原因的:

時機

PostScript 首次出現在 Apple Laserwriter 印表機中, Laserwriter 是 Macintosh 的一個完美搭擋, 而 Macintosh 系統則是導致 80 年代桌上排版革命的主要原因.

它與裝置無關 **(device-independant)**

PostScript 程式可以在畫素螢幕, 向量螢幕, 傳眞機, 或是幾乎任何一種列印機制產生輸出結果, 而不需要改變原來的程式. PostScript 在任何 PostScript 裝置下所產生的輸出結果看起來都一樣, 至少在裝置的能力限制之內是如此. 在 PDF 出現之前, 人們是以 PostScript 格式在線上交換複雜的文件. 而這個標準沒能全面成功的惟一原因就在於 Windows 機器上通常並沒有安裝 POstScript 預覽器, 於是 Adobe 指定了 PostScript 中的超連結(hyperlinks) 和壓縮法(compression), 把結果稱之爲 PDF, 並分送它的預覽器, 也替他們的文件產生工具 (ghostscript 的 ps2pdf 和 pdf2ps 兩個程式也提供了這些功能) 開發出一個新的市場.

它是一個眞正的程式語言

PostScript 是一個完整的程式語言; 你可以用它撰寫軟體來做大部份的事情. 這讓你可以在程式的開頭定義副常式, 當你需要在文件中重複做些複雜效果的時候, 像是在背景中放上商標圖案或是一個大大的 "DRAFT" 字樣的時候, 這個特點便很有用了.

它是開放的標準

PostScript 的完整規格都在一系列人人都可取得的書中 (你可以在隨便一家好的書店中找到). 雖然 Adobe 發明了它而且提供了主要的商業實作, 其它的廠商像是 Aladdin 也可以做出自己獨立開發的實作.

4.2 Non-PostScript

如果預算不允許買一部 PostScript 印表機, 你可以買任何 Ghostscript 支援的印表機. GhostScript 是一個免費的 PostScript 解譯器, 可以用來模擬實際的印表機支援. *Ghostscript* 的首頁 <http://www.cs.wisc.edu/~ghost/> 中有支援的印表機清單, 以及新的和開發中的驅動程式的最新消息. 注意一點就是這個網頁列出的是最新版的 Ghostscript 所支援的印表機, 而大部份的 Linux 發行套件因爲版權的關係只能出貨較舊的 Ghostscript 版本. 幸運的是在每一個發行套件的捐贈區(contrib area) 中通常都有先包裝好的目前版本. 請你將成功或失敗的經驗如同網頁中所要求的告訴他們以幫忙改進這個網頁.

Adobe 公司有一種新的印表機語言叫做 "PrintGear". 我想它是具有 PostScript 觀念但與其不相容的一種大幅簡化的二進位語言. 而我也沒聽說 ghostscript 支援它. 但是有些 PrintGear 印表機似乎也支援其它語言如 PCL , 而這種印表機便可以在 Linux 下使用 (假如 PCL 是在印表機中實作而不是在 Windows 驅動程式中).

4.3 那些印表機可以用?

如果你打算買一部印表機, 你可以到一些地方去看看它是否被支援. 共同合作維護的 Printing HOWTO 印表機 資料庫 <http://www.picante.com/~gtaylor/pht/printer_list.cgi> 就是希望能提供一個簡單易懂的 Linux 印表機支援狀態的列表. 簡單地將之摘要如後; 切記到線上查詢更多的細節以及該使用那個驅動程式的資料.

Ghostscript 的 相容印表機網頁 <http://www.cs.wisc.edu/~ghost/printer.html> 列出了一些支援的印表機, 同時也連結到其它的網頁. 另外 Dejanews 也有數百個"能用"和"不能用"的回報. 試試這三個地方, 在你完成以後, 請確認在 資料庫 <http://www.picante.com/~gtaylor/pht/printer_list.cgi> 中有你印表機的正確資料, 以便以後這份文件可以正確地列出你的印表機.

4.3.1 印表機相容性列表

這一節是線上版本的摘要. 線上版本包含了基本規格, 注意事項, 到驅動程式資訊的連結, 使用者維護的文件, 製造商的網頁, 諸如此類的資料. 這個列表的線上版本還是互動式的; 人們可以隨時增加印表機, 所以記得去查一下. 最後, 如果你的印表機沒有被列出來, 請把它加進去.

這裏把印表機分爲四類:

完美

這一類的印表機工作起來完美極了 - 你可以用到印表機所有的功能, 包括顏色, 所有的解析度, 等等. 在少數的情況下, 印表機具有未公開的"解析度加強"模式, 若是這種模式無法使用, 我們仍將它列爲完美; 通常這種列印品質的差別不大, 所以不需要擔心.

良好

你可以列印, 各方面的情況都不錯, 但是可能會有一些在列印或其它功能上的小限制.

勉強

你可以列印, 但是沒有辦法做彩色列印或是只能用很差的解析度. 請參閱線上列表的注意事項一欄以得知相關的限制.

悲慘

你根本沒辦法列印; 通常這是因爲缺乏驅動程式, 而且/或者是缺乏如何撰寫驅動程式的文件

由於這兒的內容是由大家所提供的, 所以沒法保證一定是正確的. 但是這應該很容易由驅動程式網頁和製造商的網站得到 corroborate. In all cases, since this information is provided by dozens of people, none of it is guaranteed to be correct. It should, however, be easy to corroborate from the driver web pages and manufacturer web sites.

廢話不多說, 下面是印表機相容性列表:

Brother

完美

HL-10V, HL-660, HL-720, HL-760.

良好

HL-1050.

Canon

完美

BJ-10e, BJ-20, BJ-200, BJC-210, BJC-240, BJC-250, BJC-610, BJC-620, BJC-70, BJC-800, LBP-8II, LIPS-III.

良好

BJ-300, BJC-4000, BJC-4100, BJC-4200, BJC-4300, BJC-4400, BJC-7000, BJC-7004.

勉強

BJC-4550, MultiPASS C2500, MultiPASS C3500.

悲慘

BJC-5000, LBP-460, LBP-660.

Epson

完美

ActionLaser 1100, LP 8000, LQ 850, Stylus Color, Stylus Color 400, Stylus Color 500, Stylus Color 600, Stylus Color 640, Stylus Color 850, Stylus Color II, Stylus Color IIs, Stylus Pro XL.

良好

Stylus Color 800, Stylus Photo 750.

勉強

Stylus Color 740.

HP

完美

2000Cse, 2500C, DesignJet 650C, DeskJet 1200C, DeskJet 1600C, DeskJet 1600Cm, DeskJet 400, DeskJet 420C, DeskJet 500, DeskJet 550C, DeskJet 600, DeskJet 660Cse, DeskJet 690C, DeskJet 850C, DeskJet 855C, DeskJet 870, DeskJet 870Cxi, DeskJet 890, LaserJet 1100, LaserJet 2100, LaserJet 2100M, LaserJet 4000N, LaserJet 4L, LaserJet 5, LaserJet 5L, LaserJet 5MP, LaserJet 6L, LaserJet 6MP, LaserJet 8000, LaserJet 8100, Laserjet 5000, Mopier 320, PaintJet XL300.

良好

HP 660C.

勉強

DeskJet 1000C, DeskJet 670C, DeskJet 710, DeskJet 720C, DeskJet 722C, DeskJet 820C.

悲慘

LaserJet 3100.

IBM

完美

Jetprinter 3852.

Lexmark

完美

Optra Color 1200, Optra Color 1275, Optra Color 40, Optra Color 45, Optra E, Optra E+, Optra Ep, Optra S 1250.

勉強

1020 Business, 3000, 5700, 7000.

悲慘

1000, 1020, 1100, 2030, 2050, 2070, 5000, 7200, Winwriter 100, Winwriter 150c, Winwriter 200.

Minolta

完美

PagePro 6, PagePro 8.

NEC

完美

P2X.

勉強

SuperScript 100C, SuperScript 1260, SuperScript 150C, SuperScript 650C, SuperScript 750C, SuperScript 860, SuperScript 870.

悲慘

SuperScript 660i, SuperScript 660plus.

Okidata

完美

OL 410e, OL 610e/PS, OL 810e/PS, Okipage 6e, Okipage 6ex, Okipage 8c.

良好

Okipage 4w.

悲慘

Okipage 8w, okijet 2010.

Olivetti

完美

JP350S.

Panasonic

完美

KX-P1123, KX-P4440, KX-P5400, KX-P8420, KX-P8475.

勉強

KX-P6500.

悲慘

KX-P6100, KX-P8410.

QMS

完美

2425 Turbo EX.

Ricoh

完美

4801, 6000.

Xerox

完美

DocuPrint C55, DocuPrint N17, DocuPrint N32.

5 印表連線作業軟體(spooling software) 呢?

直到最近, Linux 的使用者的選擇很簡單 - 每個人跑的都是古老的從 BSD 的 Net-2 程式碼移過來的 lpd. 即使在今天, 大部份廠商仍以這個軟體出貨. 但是這個情況已經開始在改變. SVR4 一派的系統, 包括昇陽的 Solaris 在內所附的是個完全不同的印表機連線作業套件, 這個套件以 lpsched 爲中心. 另外還有跡象顯示有些 Linux 廠商會改成提供新得多而且免費的 LPRng. LPRng 的程式碼遠比舊有的 lpd 來的有規劃, 在大量安裝時(註: 超過一台印表機, 序列式印表機, 或任何非 lpd 的網路印表機)也較爲容易管理. 它甚至可以眞的稱得上安全 - 沒有 SUID 的執行檔, 還支援透過 PGP 或 Kerberos 做使用者驗証.

雖然有這些新選擇, 目前 lpd 對大部份的 Linux 使用者而言應該是足夠了. 雖然它不是最新潮的系統, 但只要設定好就沒問題, 而且它廣爲人們所熟悉, 同時很多的 Unix 書籍也提供了大量的說明文件.

如果你想知道更多有關 LPRng 的資訊, 請到 *LPRng* 的網頁 `<http://www.astart.com/lprng/LPRng.html>`. 在未來這份 HOWTO 中將會包含使用 LPRng 以及使用普通 lpd 的相關訊息.

6 基本運作方式

爲了讓列印功能能夠正常運作, 你需要瞭解 lpd 系統是如何運作的.

Lpd 是 Line Printer Daemon 的縮寫, 根據上下文指的可以是監控程式(the daemon) 本身或是執行印表機連線作業的全套程式. 包括了有:

lpd

印表機連線作業監控程式(The spooling daemon). 有一個負責控制所有事情, 另外針對每一部列印中的印表機也會執行一份.

lpr

使用者的列印指令. Lpr 會聯絡 lpd 並將列印工作送進列印暫存器.

lpq

列出佇列中等待列印的工作.

lpc

Lpd 系統控制指令. 你可以用 lpc 來控制佇列, 如停止, 開始, 調整順序等等.

lprm

`lprm` 可以從佇列中移除一件列印工作.

那這一切是如何湊在一起的呢? 當系統開機時會執行 lpd. 它會檢查 */etc/printcap* 中有那些印表機要設立列印暫存器. 每當有使用者執行 lpr 時, lpr 會透過 */dev/printer* 的插座(socket) 找到 lpd, 並將要列印的檔案和一些關於列印的人以及列印方法的資訊傳送給 lpd. 然後 lpd 再將檔案送到的適當印表機.

lp 系統設計的時候, 大部份的印表機都是列式印表機(line printer) - 也就是說, 大部份的使用者是列印純文字格式(plain ascii). 結果就是我們還得做一些額外的命令稿處理(scripting) 才能使 lpd 適用於現今的列印工作, 通常是 PostScript, 或是文字格式, 或是 dvi, ...

7 基本系統設定

7.1 傳統的 lpd 設定

最基本的 lpd 設定只能讓系統佇列檔案然後列印, 而不管你的印表機能否認得這些檔案, 可能也無法讓你產生吸引人的報表. 但無論如何, 這是跨向理解的第一步. 所以讓我們看下去.

基本上, 要增加一個列印佇列, 你必須在 */etc/printcap* 中加一筆記錄, 以及在 */var/spool/lpd* 之下建立一個列印暫存器的工作目錄(spool directory).

/etc/printcap 中的記錄看起來像這個樣子:

```
# LOCAL djet500
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :sh:
```

這個例子定義了一個名爲 *lp*, *dj*, 或是 *deskjet* 的列印暫存器, 暫存器的工作目錄在 */var/spool/lpd/dj*, 沒有單一工作大小的上限, 列印的裝置是 */dev/lp0*, 而且不會在每一個工作之前加上標題頁 (有著列印人的姓名之類的資料).

現在就去看看 printcap. 的線上手冊.

上面這個例子看起來非常簡單, 但是有一個問題 - 除非 DeskJet 500 能看懂我送出去的檔案, 否則便會印出亂碼. 舉例來說, 假如送一個傳統的 Unix 文字檔案給 deskjet 會導致新行字元(newline) 被逐字列印, 而得下面的結果:

```
This is line one.
                 This is line two.
                                  This is line three.
```

同樣的, 列印一個 PostScript 檔到這個列印暫存器會得到同樣階梯狀的 PostScript 命令列表, 而不是我們需要的結果.

很顯然我們還需要些什麼, 這就是 filtering 的目的. 看過 printcap 線上手冊的比較細心的讀者可能已經注意到 *if* 及 *of* 這兩個屬性. 其中的 *if*, input filter 就是我們現在所需要的.

假使我們寫個短短的 shell script 叫 *filter* 使之在新行字元之前加上回車字元(carriage return), 就可以消除階梯狀. 所以我們必須在前面的 printcap 的記錄中加上 *if* 這一行:

```
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/dj/filter:\
        :sh:
```

這個短短的 filter script 可以是:

```
#!perl
# The above line should really have the whole path to perl
# This script must be executable: chmod 755 filter
while(<STDIN>){chop $_; print "$_\r\n";};
# You might also want to end with a form feed: print "\f";
```

假如我們照上面的方法, 我們就會得到一個能正常印出普通 Unix 文字檔的列印暫存器. (還有其他四百萬種更好的方法來寫這個命令稿, 但是這個很適合舉例, 你可以自己寫一個更有效率的.)

剩下的唯一問題是只能印出純文字檔實在是不夠好 - 如果能夠印出 PostScript 或是其它格式化的或是圖形輸出就更棒了. 這可以做的到, 也很容易做到. 方法就是延伸前面的 filter. 如果你寫出一個能接受任意檔案格式, 然後針對每種情況都可以產生 DeskJet 所要的資料的 filter, 那你就有了一個智慧型的列印暫存器.

這種 filter 稱之為 *magic* filter. 你無需自己寫一個, 除非你要印的東西很奇怪. 網路上已經有很多寫好的, APS Filter 就是非常棒的一個. 也說不定你的 Linux 發行版本(distribution) 已經包括了一個印表機設定工具使得這一切都很容易.

7.2 檔案權限

應大眾要求, 我把我系統上一些相關檔案的權限列出. 還有其它更好的設定方法, 最好是只將執行檔設為 SGID 而不要設為 SUID root, 但這是我系統預設的設定, 而且運作正常.(老實說, 如果你的廠商連可以運作的 lpd 都無法提供, 你就有苦頭吃了.)

```
-r-sr-sr-x   1 root     lp    /usr/bin/lpr*
-r-sr-sr-x   1 root     lp    /usr/bin/lprm*
-rwxr--r--   1 root     root  /usr/sbin/lpd*
-r-xr-sr-x   1 root     lp    /usr/sbin/lpc*
drwxrwxr-x   4 root     lp    /var/spool/lpd/
drwxr-xr-x   2 root     lp    /var/spool/lpd/lp/
```

Lpd 目前必須以 root 身份執行才能連接到低號(連接埠號低於 1024)的 lp 服務埠. 可能應該改成在連接後將使用者身份變成 lp 或其他身份, 但是我不認為現在有這樣做. 令人有些失望.

8 取得列印軟體

許多預先寫好的 filter 套件 (以及其它與印表有關的軟體) 在 *Metalab* <ftp://metalab.unc.edu/pub/Linux/system/printing/> 都有提供. 像是 psutils, a2ps, mpage, dvitodvi, flpr 等工具都可以在那兒找得到.

有一陣子, 很多套件都想讓設定印表機簡單一點. 這些套件可能都還在, 但是其中最新最好就屬 Andreas Klemm 的 APS Filter 套件了. 這個套件提供了一個選單式的 printcap 設定程式, 而且幾乎可以處理任何你所能想到的輸入格式. 假使你的廠商並未提供一組好的印表機設定工具, 選 APS Filter 就對了.

9 廠商的解決方案

理論上, 這一節是不完整的. 請別客氣來告訴我你最喜好的發行版本(distribution) 的相關細節.

9.1 Red Hat

Red Hat 提供一個 GUI 界面的印表機管理程式 (在 control panel 裏), 可以讓你加上網路印表機或連接本地裝置的印表機. 它讓你選擇 ghostscript 支援的印表機種類和所使用的 Unix 裝置檔案, 然後會在 */etc/printcap* 中設定列印佇列, 再寫入一個簡短的利用 gs 和 *nenscript* 以處理 PostScript 和 ascii 的 magic filter. 這個方法還不錯, 對一般的情況而言, 也很容易設定.

但如果你的印表機不在 Red Hat 標準配備的 Ghostscript(GNU 版而不是 Aladdin, 支援的印表機種類較少) 支援之列就不太妙了. 假如你發現用 Red Hat 所附的軟體無法正常列印的時候, 請參閱上面的印表機相容性列表 (或是到 線上資料庫 <http://www.picante.com/~gtaylor/pht/printer_list.cgi>). 如果 Red Hat 的工具不支援你的印表機的話, 你可能需要安裝免費版的 Aladdin Ghostscript, 如果你使用 apsfilter 套件可能會更好. apsfilter 套件能處理所有新近 Ghostscript 支援的印表機.

9.2 Debian

Debian 讓你選擇普通 lpd 或 LPRng; LPRng 可能是比較好的選擇. 我相信 Debian 也提供印表機設定工具的選擇; apsfilter 5 或之後的版本應該是最適合你的, 因為這一版增加了對 LPRng 和 Ghostscript 統一驅動程式(uniprint driver scheme) 的支援.

9.3 其它的發行版本(Other Distributions)

請將其它發行版本的處理方法的訊息寄給我!

10 Ghostscript.

Ghostscript 是個對 Linux 列印有著極大意義的程式. 大部份在 Unix 下列印的程式都會產生 PostScript, 而印表機的 PostScript 處理能力通常是價值 100 美元的選購配備. 但是 ghostscript 卻是免費的, 而且它能將 PostScript 轉成你的印表機所用的語言. 當它和你的 `lpd` input filter 結合之後, 你就有了一部虛擬的 PostScript 印表機, 而且能夠大幅簡化你的生活.

Ghostscript 有兩個版本. 商業版稱爲 Aladdin Ghostscript, 個人可以自由使用, 但是不能隨著商業用的 Linux 發行版本散佈. Aladdin Ghostscript 通常比免費的 ghostscript 領先一年左右; 舉例來說, 它支援許多免費版尚未支援的彩色噴墨機.

免費版的 ghostscript 爲 GNU Ghostscript, 其實就是較舊版的 Aladdin Ghostscript. (Aladdin 的這種安排值得我們讚賞; 應該有更多無法完全以 GPL 來散佈原始碼的軟體廠商以這種方式支持自由軟體 (free software).

無論你用 *gs* 做什麼, 記得不要讓它能夠存取檔案 (用 `-dSAFER` 選項). PostScript 是個具有完整功能的語言, 而一個惡意的 PostScript 程式可以讓你吃足苦頭.

再談到 PDF, Adobe 的 Portable Document Format 實際上就是把 PostScript 更組織化, 加上一些擴充功能, 經過壓縮的檔案格式. Ghostscript 可以像處理 PostScript 般地處理 PDF. 所以你可能是你週遭第一個擁有具有處理 PDF 能力的印表機的人.

10.1 啓動 ghostscript

典型的作法是在你的 magic filter(假如你的廠商沒有提供一個適合你用的, 我建議 apsfilter) 中使用 ghostscript, 但是如果爲了除錯, 直接執行通常會比較方便些.

`gs -help` 會簡短地列出有用的選項以及所提供的驅動程式. (注意這兒所列出的僅是編譯進去的驅動程式, 而並非所有的驅動程式).

你可以用下面的方式執行 gs 來測試: `gs` *`options`* `-q -dSAFER -sOutputFile=/dev/lp1 test.ps`.

10.2 調整 ghostscript 的輸出結果

假如 gs 的輸出不能讓人滿意, 還有一些辦法.(事實上, 你可以做任何事, 因爲你可以取得原始碼).

10.2.1 輸出的位置及大小

位置, 大小, 還有影像在頁面上的視覺比例是由 ghostscript 中的印表機驅動程式所控制的. 如果你發現列印出來太短, 太長或是大了兩倍, 你可能會想要看一下驅動程式的原始模組並且調整任何所能想到的參數. 不幸的是每一個驅動程式都有不同的方式, 所以我也無法告訴你如何調整, 還好大部份的驅動程式都有還算清楚的註解.

10.2.2 Gamma, dotsizes, 等等

大部份的非雷射印表機因爲列印的點比較大, 故會導致印出來的圖案太黑. 假使你遇到這個問題, 你應該使用自己的轉換函數. 只要在 ghostscript 的 lib-dir 下建立如下的檔案, 並在呼叫 gs 的時候將這個檔名加在要列印的檔案之前就可以了. 你可能會需要調整一下實際的值才能適合你的印表機. 較小的值會

導致較亮的列印效果. 特別是當你的印表機使用 Floyd-Steinberg 演算法來混色 (rasterize color) 可能是比較好的選擇.

```
---8<---- gamma.ps ----8<---
%!
%transfer functions for cyan magenta yellow black
{0.3 exp} {0.3 exp} {0.3 exp} {0.3 exp} setcolortransfer
---8<------------------8<---
```

你也可以調整這些值來補救印表機顏色上的問題. 假如你要這麼做, 建議你可以用 ghostscript 內附的 colorcir.ps (在 examples/ 目錄下) 做爲測試頁.

很多較新型的彩色噴墨印表機的驅動程式具備命令列選項, 或是有不同的 upp 驅動程式檔來實做 gamma 及其它改變以讓印表機適應不同類型的紙張. 在你用 Postscript 來修正問題之前, 應該先從這方面著手.

11 如何列印到網路印表機

Lpd 的一個特性便是可以透過網路將報表列印到實體連接在其它機器上的印表機. 只要仔細地組合 filter 命令稿和相關的工具程式, 你就可以讓 lpr 得以透通地 (transparently) 列印到各種網路上的印表機.

11.1 列印至 Unix/lpd

要讓遠端的機器能夠使用你的印表機列印, 你必須將遠端機器列名在 */etc/hosts.equiv* 或是 */etc/hosts.lpd* 檔案之中. (要注意 *hosts.equiv* 還有其它很多作用; 如果你要將一部機器列在這兒, 請確定你知道你在做什麼). 你也可以利用 *rs* 屬性來限定其它機器上的某些特定使用者才能使用你的印表機. 請看 *lpd* 的線上手冊以取得這方面的訊息.

11.1.1 用 lpd

要列印到另一部機器, 你要在 */etc/printcap* 中加上這樣的記錄,

```
# REMOTE djet500
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :rm=machine.out.there.com:\
        :rp=printername:\
        :lp=/dev/null:\
        :sh:
```

請注意在本地的機器還是要有一個由 `lpd` 管理的列印暫存器目錄. 如果遠端的機器離線或是忙不過來, 列印工作會留在本地的暫存區中直到送出去爲止.

11.1.2 用 rlpr

你也可以用 *rlpr* 將列印工作直接送到遠端機器的佇列中, 省下設定 lpd 的麻煩. 假如你偶爾才要列印到其它不同的印表機, 這個方法就最有用. 引一段 *rlpr* 的聲明:

rlpr 利用 TCP/IP 將列印工作送到網路上任何位置的 lpd 伺服器.

與 lpr 不同的是, 你列印來源端的機器 *不需要* 明確地知道遠端的印表機, (例如透過 */etc/printcap*) 因此有相當大的彈性, 管理的負擔也比較輕.

rlpr 可以用在任何使用傳統 lpr 的地方, 而且和傳統的 BSD lpr 相容.

rlpr 最大的好處在於可以 *從任何地方列印到任何地方* 而不用管列印來源端的系統設定. 它可以像傳統的 lpr 做爲一個 filter, 而讓在遠端機器上執行的客户端像 netscape, xemacs 等能夠輕易地列印報表到你本地的機器.

你可以在 *Metalab* <ftp://metalab.unc.edu/pub/Linux/system/printing/> 找到 rlpr.

11.2 列印到 Win95, WinNT, LanManager, 或 Samba 上的印表機

在 Printing to Windows mini-HOWTO 中有比這兒更多的資訊.

透過 *smbclient* 這個程式 (samba 套件的一部份) 可以將 lpd 佇列導向到架構在 TCP/IP 之上的 SMB 列印服務. Samba 包含了一個命令稿 `smbprint` 來做這件事. 簡單地說, 替這樣的印表機在列印暫存器工作目錄 (spool directory) 中加上設定檔, 然後將 *if* 設爲 `smbprint` 命令稿.

在 */etc/printcap* 中的記錄看起來應該像這個樣子:

```
lp|remote-smbprinter:\
    :lp=/dev/null:sh:\
    :sd=/var/spool/lpd/lp:\
    :if=/usr/local/sbin/smbprint:
```

你應該閱讀一下 `smbprint` 命令稿中的說明以得到更多相關的訊息.

你也可以不管 `lpd`, 直接用 `smbclient` 將檔案送到 SMB 列印服務. 請參照線上手册.

11.3 列印到 NetWare 上的印表機

ncpfs 套件中包含了一個名爲 `nprint` 的工具程式, 它提供如同 `smbprint` 一般的功能, 只不過適用的對象是 Netware. 你可以在 *Metalab* 取得 ncpfs. 下面是從 0.16 版的 LSM 中節錄出來的:

> With ncpfs you can mount volumes of your netware server under Linux. You can also print to netware print queues and spool netware print queues to the Linux printing system. You need kernel 1.2.x or 1.3.54 and above. ncpfs does NOT work with any 1.3.x kernel below 1.3.54. 利用 ncpfs, 你可以在 Linux 下掛上你的 netware 伺服器上的目錄. 你也可以列印到 netware 的列印佇列以及將 netware 的列印佇列排存(spool) 到 Linux 的列印系統. 你需要 1.2.x 或 1.3.54 以上(含 1.3.54) 的核心. ncpfs 無法在 1.3.54 以前的任何一版 1.3.x 核心下運作.

要使 `nprint` 透過 lpd 運作, 你要寫個小小的 shell script 將 stdin 列印到 Netware 上的印表機, 並設成 lpd 列印佇列的 *if*. 你可能會有:

```
sub2|remote-NWprinter:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/sub2:\
        :if=/var/spool/lpd/nprint-script:
```

nprint-script 看起來可能大致上像:

```
#! /bin/sh
# You should try the guest account with no password first!
/usr/local/bin/nprint -S net -U name -P passwd -q printq-name -
```

11.4 列印到 EtherTalk (Apple) 上的印表機

netatalk 套件包含了像 nprint 和 smbclient 的東西. 其他人已經寫下了使用 Apple 網路列印的程序, 遠比我所能寫的好得多; 請參閱 *Linux Netatalk-HOWTO* <http://thehamptons.com/anders/netatalk/>.

11.5 列印到 HP 或其它的以太網路(ethernet) 印表機

有些 HP 和其它廠牌的印表機提供了可以直接用 lpd 列印的以太網路介面. 你應該遵照印表機或其網路介面所附的指示, 但是通常這種印表機上 "跑" 的是 lpd, 而且提供一到多個你可以使用的佇列. 舉例來說, 一部 HP 可能可以用這樣的 printcap:

```
lj-5|remote-hplj:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/lj-5:\
        :rm=printer.name.com:rp=raw:
```

具有 Jet Direct 介面的 HP Laserjet 印表機通常支援兩個內建的 lpd 佇列 - "raw" 接受 PCL (可能也接受 Postscript), 以及 "text" 接受單純的 ascii (而且會自動地克服階梯效應). 如果你有的是 JetDirect Plus3 三埠的介面, 佇列的名稱則爲 "raw1", "text2" 等等.

注意 ISS 公司已經找到某一類的過量攻擊(denial of service attacks) 可以導致 HP Jetdirect 界面當掉. 大部份是在 98 年的秋天開始被發現的.

在一個大型的環境, 特別是個有些印表機不支援 PostScript 的大型環境中, 架設一部專門的列印伺服器, 讓所有的機器都列印到這部伺服器並讓所有的 ghostscript 工作都在這部伺服器上面執行可能會有用.

這樣子也可以讓你的 Linux 做爲印表機的排存伺服器(spool server), 如此一來你的網路使用者便可以快一點完成他們的列印工作然後繼續手邊的事情, 而不必等待印表機印完先前別人所送出的報表. 如果你有無法調整的舊型 HP Laserjets; 也可以這樣做以降低印表機壅塞的機率.

要這樣做, 先在你的 linux 機器上設定一個列印佇列指到配有以太網路介面 HP LJ (如前所述). 現在將所有在你區域網路上的客户端設定指到這個 Linux 佇列 (如前面範例中的 lj-5).

有些 HP 網路印表機似乎不理會客户端送出的標題頁(banner page) 設定. 你可以關掉它們內部自行產生的標題頁, 用 telnet 連接到印表機, 按兩下 return 鍵, 打入 "banner: 0", 接著打入 "quit". 還有其它可以調整的設定也是用這個方式; 打入 "?" 可以看到它們的表列.

你可以用 HP 的 *WebJet* <http://www.hp.com/go/webjetadmin> 軟體來控制全部的設定. 這套軟體是以 daemon 的方式執行, 接受在指定埠號上的 http 連線需求(http request). 它提供 http 表格(form) 及 Java applets 來控制網路上的 HP 印表機. 理論上, 它也可以控制 Unix 的印表機佇列, 但是這得使用極不安全的的 rexec 服務. 我不建議使用這個功能.

11.5.1 列印到比較舊的 HP 印表機

有些印表機 (或是將印表機連上網路的裝置) 只支援很陽春的 TCP 連線. 在這一類中較著名的便是早期的 JetDirect (包括一些 JetDirectEx) 介面卡. 基本上, 要列印到這類的印表機, 你必須建立到印表機上某個特定埠 (典型是在 9100, 或是三埠機種的 9100, 9101 和 9102) 的 TCP 連接, 然後將你的報表送進去. LPRng (如果你正在讀這一段, 你眞的該改用它) 內建支援將列印工作送到任意的 TCP 埠, 但是 BSD 的 lpd 就不是這麼簡單了. 我們可以用 Perl 來實做, 或是用其它的方式:

```
#!/usr/bin/perl
# Thanks to Dan McLaughlin for writing the original version of this
# script (And to Jim W. Jones for sitting next to Dan when writing me
# for help ;)

$fileName = @ARGV[0];

open(IN,"$fileName") || die "Can't open file $fileName";

$dpi300     = "\x1B*t300R";
$dosCr      = "\x1B&k3G";
$ends = "\x0A";

$port =  9100 unless $port;
$them = "bach.sr.hp.com" unless $them;

$AF_INET = 2;
$SOCK_STREAM = 1;
$SIG{'INT'} = 'dokill';
$sockaddr = 'S n a4 x8';

chop($hostname = `hostname`);
($name,$aliases,$proto) = getprotobyname('tcp');
($name,$aliases,$port) = getservbyname($port,'tcp')
    unless $port =~ /^\d+$/;;
($name,$aliases,$type,$len,$thisaddr) =
        gethostbyname($hostname);
($name,$aliases,$type,$len,$thataddr) = gethostbyname($them);
$this = pack($sockaddr, $AF_INET, 0, $thisaddr);
$that = pack($sockaddr, $AF_INET, $port, $thataddr);

if (socket(S, $AF_INET, $SOCK_STREAM, $proto)) {
#    print "socket ok\n";
}
else {
    die $!;
}
# Give the socket an address.
if (bind(S, $this)) {
#    print "bind ok\n";
}
else {
    die $!;
```

```
}

# Call up the server.

if (connect(S,$that)) {
#    print "connect ok\n";
}
else {
    die $!;
}

# Set socket to be command buffered.

select(S); $| = 1; select(STDOUT);

#    print S "@PJL ECHO Hi $hostname! $ends";
#    print S "@PJL OPMSG DISPLAY=\"Job $whoami\" $ends";
#    print S $dpi300;

# Avoid deadlock by forking.

if($child = fork) {
    print S $dosCr;
    print S $TimesNewR;

    while (<IN>) {
        print S;
    }
    sleep 3;
    do dokill();
} else {
    while(<S>) {
        print;
    }
}

sub dokill {
    kill 9,$child if $child;
}
```

11.6 替遠端印表機執行 *if*

lpd 一個不尋常的地方在於如果是遠端印表機, *if* 不會被執行. 如果你發現你需要執行 *if*, 你可以設定雙重佇列(double queue) 並重新佇列報表. 舉個例子, 看一下這個 *printcap*:

```
lj-5:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/lj-5:\
        :if=/usr/lib/lpd/filter-lj-5:
lj-5-remote:lp=/dev/null:sh:rm=printer.name.com:\
```

```
          :rp=raw:sd=/var/spool/lpd/lj-5-raw:
```

和相關的 `filter-lj-5` 命令稿(script):

```
#!/bin/sh
gs <options> -q -dSAFER -sOutputFile=- - | \
        lpr -Plj-5-remote -U$5
```

只有在 lpr 以監控程式(daemon) 模式執行的時候, *-U* 選項才有作用, 這個選項會正確地設定第二重佇列中的送件人姓名. 你可能應該用更可靠的方法取得送件人姓名, 因爲有些時候它並不是第五個參數. 請參照 printcap 的線上手冊.

11.7 從 Windows 列印

利用 SAMBA 套件可以直接支援以 SMB 協定從 Windows (或許 OS/2) 客户端列印到 Linux 伺服器, 這個套件同時也支援將你的 Linux 檔案系統分享給 Windows 客户端.

Samba 包含了相當完整的文件, 也有一份很不錯的 Samba FAQ 涵蓋到這方面的內容. 你可以在 Linux 上設置 magic filter 然後以 PostScript 列印, 或是在每一部 Windows 機器上安裝印表機專用的驅動程式然後爲他們設定一個不用任何 filter 的佇列. 使用 Windows 驅動程式有些時候會有比較好的輸出, 但是如果有很多部 Windows 機器, 管理者就比較麻煩. 所以先試試看 PostScript 吧!

11.8 從 Apple 列印

Netatalk 支援由 Apple 客户端透過 EtherTalk 列印. 請參照 *Netatalk HOWTO Page* `<http://thehamptons.com/anders/netatalk/>` 以取得更進一步的資訊.

11.9 從 Netware 列印

ncpfs 套件含有一個叫作 pserver 的監控程式(daemon) 可以提供服務給 NetWare 的列印佇列. 據我所瞭解, 這個系統必須搭配 Bindery-based 的 NetWare, 像是 2.x, 3.x, 或是使用 bindery 存取的 4.x 版本.

要取得更進一步關於 ncpfs 和其 pserver 程式的資訊, 請參照 *ncpfs* 的 *FTP* 站 `<ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/>`.

12 Windows 專用印表機

如同稍早之前我所討論的, 有些印表機原本就不被支援, 因爲它們並非使用一般的印表機語言, 而是利用電腦的 CPU 來描繪出位元組列(bitmap), 然後再以固定的速率將之導到(pipe to) 印表機. 少數這類的印表機也使用普通的印表機語言, 像PCL, 但是通常都沒有. 有些 (非常低階的) 印表機甚至並非使用平行埠連接, 而是靠廠商的驅動程式來模擬硬體的行爲 (最重要的流量控制(flow control)).

假如你發現你陷在這種惡劣的情況, 無論如何, 總還有一些可能的解決之道.

12.1 Ghostscript Windows 導向器(redirector)

現在有一個 Windows 的印表機驅動程式可以用 (叫做 mswinpr2), 它可以在列印出報表之前先將其送到 Ghostscript 處理. (有點像在 Unix LPD 中的 if filter). 也有一個新的 Ghostscript 驅動程式可以用 Windows GDI 呼叫來列印. 把它們兜在一塊兒, 這樣應該可以讓 Windows 機器用廠商的驅動程式列印 PostScript 到一部 Windows 專用的印表機.

如果你成功了, 你可以依照前面讓 Linux 透過網路列印到 Windows 印表機的說明, 讓 Unix (以及其它的 Windows, Mac 等) 主機列印到你的印表機.

話雖如此, 我從來沒試過, 因爲我沒有 Windows 專用的印表機, 所以你的評價可能不同. 所有這兒所牽涉到的軟體應該都可以從 Ghostscript 的首頁 (請看這份文件在 Ghostscript 一節中的連結) 找到.

12.2 pbm2ppa 程式

有些 HP 的印表機使用 "Printing Performance Architecture" ("我們太便宜了, 所以沒有 PCL" 的行銷術語). 要支援這種印表機, 可以透過使用 Tim Norman 所寫的 pbm2ppa 翻譯器(translator) 的這種迂迴的方式. 基本上, 你使用 ghostscript 將 PostScript 描繪成 pbm 格式的位元組列圖案(bitmapped image), 然後用 pbm2ppa 將其翻譯成可以直接傾印(dump) 的印表機專屬的 ppa 格式的位元組列(bitmap). 這個程式現在可能也以 ghostscript 驅動程式的型式出現.

ppa 軟體可以由 *ppa* 的首頁 `<http://www.rpi.edu/~normat/technical/ppa/>` 取得. pbm2ppa 支援 HP 720, 820, 和 1000 的一些機種; 請閱讀套件中所附的文件以取得有關 ppa 印表機支援的更進一步資訊.

12.3 pbm2l7k 程式

大部份便宜的 Lexmark 噴墨印表機使用一種專用的語言而成了 Windows 專用的印表機. 還好, Henryk Paluch 寫了一個程式可以在 Lexmark 7000 上做黑白列印. 他有可能能夠找出彩色列印的方法並且支援其它的 Lexmark 噴墨印表機. 請參照 這個網站 `<http://bimbo.fjfi.cvut.cz/~paluch/l7kdriver/>`以取得更多資訊.

13 如何列印到傳眞機

13.1 使用傳眞數據機(faxmodem)

外面有很多的傳眞程式可以讓你傳眞及接收文件. 最複雜的其中一個就是 Sam Leffler 的 *HylaFax*, 可以在 `ftp.sgi.com` 取得. 它有各式各樣的功能, 從使用多部數據機到廣播傳送(broadcasting).

SuSE 出了一個 Java 版的 HylaFax 客户端應該可以在任何的 Java 平台上運作 (包括 Windows 和 Linux). 大部份的平台上也有非 Java 版的傳眞客户端; Linux 幾乎可以完全地處理你的網路傳眞的需要.

對大部份的 Linux 機器而言, 還有一個較好的選擇, 就是 *efax*, 一個送傳眞的簡單程式. 而 getty 的程式 `mgetty` 則可以接收傳眞 (在有些數據機上甚至可以接收語音郵件!).

13.2 使用遠端列印服務

有人提供一種實驗性的服務, 讓你可以用電子郵件送出你想要列印的內容, 然後從別處的傳眞機列印出來. 這個服務支援一些不錯的格式, 像是 PostScript, 雖然涵蓋的區域不夠廣泛, 仍不失爲一個

很有用的服務. 想要知道更多有關於使用遠端列印服務的訊息, 請參照 *Remote Printing WWW Site* <http://www.tpc.int/>.

14 如何產生值得列印的東西

我們可以使用的軟體眞是又多又雜. 基本上, Linux 可以執行很多種的二進位檔, 相容(執行成功) 的程度各有不同: Linux/x86, Linux/Alpha, Linux/Sparc, Linux/foo, iBCS, Win16/Win32s (利用 dosemu, 或許將來用 Wine), Mac/68k (用 Executor), 和 Java. 我將只討論原生的(native) Linux 和常見的 Unix 軟體.

就 Linux 自身而言, 選擇幾乎僅限於那些 Unix 一般所能取得的:

14.1 標示語言(Markup languages)

大部份的標示語言(markup languages) 比較適合大型的或是重覆性的計畫, 你可以用電腦來控制文字的版面設計以取得一致性.

`nroff`

這是 Unix 最早的標示語言之一. 最常見的範例 - 線上手冊(man pages) - 便是用 *roff 巨集編排而成的; 許多人信賴 nroff, 但是它的語法比所需要用到的更艱澀難懂(至少對我而言), 對新的工作而言, 它可能並不是個好選擇. 話雖如此, 知道你可以用 groff 將線上手冊直接排版成 PostScript 還是值得的. 用 `man -t foo | lpr`, 大部份的 man 指令會替你做這件事.

TeX

TeX, 和巨集套件 LaTeX, 是 Unix 上最被廣泛使用的標示語言之一. 技術性的工作通常用 LaTeX 來撰寫因爲它能大幅地簡化格式編排的問題, 而且仍然是少數能完整並良好地支援數學符號的文字處理器之一. TeX 的輸出格式爲 `dvi`, 可以用 `dvips` 或 `dvilj` 將其轉換爲 PostScript 或是 Hewlett Packard 的 PCL. 如果你要安裝 TeX 或 LaTeX, 安裝整組 teTeX 套件; 它包含了所需的一切.

SGML

Unix 和 Linux 至少有一個免費的 sgml 語法解析程式(parser) 可用; 它是 Linuxdoc-SGML 產生的文件系統的基礎. 它也可以支援其它的 DTD(文件型態定義).

HTML

有人建議説對於比較簡單的計畫, 用 HTML 撰寫, 然後用 Netscape 列印可能就足夠了. 我並不贊成, 但是你可能有不同的評價.

14.2 所見即所得(WYSIWYG) 的文書處理器

所見即所得的文書處理器再也不怕沒有了. 有好幾套辦公室套裝軟體(office suite), 其中還有一套(StarOffice) 可以讓個人免費使用.

StarOffice

一家德國公司在網路上免費散佈 Linux 版本的 StarOffice. 這個全功能的辦公室套裝軟體有著所有你想要的功能, 而且你無法抗拒它的價格. 有一份 mini-HOWTO 説明如何取得並安裝它. 它產生 PostScript 和 PCL, 所以應該可以和大部份能在 Linux 上使用的印表機一起使用. 很明顯的, 它是一套 Office 的仿製品, 而且相當的龐大臃腫.

WordPerfect

Corel 免費發行了一套 Linux 版本上的 Word Perfect 8 簡易版, 而且已經暗示一旦移植的工作完成, 他們也會發行 Corel Draw 和 Quattro Pro. 如果你有一部使用 ARM 處理器的機器, 這可能是你最好的選擇; Corel 開發出幾款使用 ARM 處理器的 Netwinder Linux 主機, 而且可以說是一定會提供 ARM Linux 版本的所有東東. 你也可以選擇購買全功能的版本及支援服務, 兩者一起或是分別購買.

Linux WordPerfect 字型與印表機 <http://www.channel1.com/users/rodsmith/wpfonts.html>首頁有關於設定 WordPerfect 使用 Ghostscript 或是內建的印表機驅動程式的資料. (內建的印表機驅動程式很明顯地和 DOS 版 WordPerfect 的驅動程式一模一樣, 萬一你的印表機的驅動程式不在 WP8 的發行套件中的話, 可以試試.)

Applix

Applix 是由 Applix 公司銷售的一個跨平臺(如各種不同版本的 Unices, Windows 和其它)的辦公室套裝軟體. Red Hat 和 SuSE 在它是唯一選擇的時候曾經銷售過; 現在則將銷售權交還給 Applix.

LyX

LyX 是個值得期待的 LaTeX 前端. 請看 *LyX* 首頁 <http://www.lyx.org/> 以取得更進一步的訊息. 有一個搭配 KDE 的 LyX 版本, 叫做 Klyx; LyX 的作者和整合 KDE 的人是同一個.

Maxwell

Maxwell 是個簡單的使用 MS RTF 格式的文書處理器, 一開始是個商業產品, 但現在則是採用 GPL 版權發行.

The Andrew User Interface System

AUIS 包括了 ez, 一個所見即所得風格的編輯器, 它有著大部份基本的文書處理功能, HTML 的能力, 以及完全地支援 MIME 格式的電子郵件和新聞群組. 不幸的是, 沒人再繼續維護 AUIS.

Koffice

KDE 計畫正在朝向全套的辦公室套裝軟體努力. 我不認爲它已經準備好讓大眾來使用. 它的文書處理器很明顯的是 LyX 的後代.

GNOME

GNOME 計畫也正在努力開發許多不同的 GNU 授權的辦公室工具. 但是還沒一個可用.

Jeff Phillips <jeff@I_RATUS.org> 使用 Caldera 的 Linux 版本的 WordPerfect 7(在 Slackware 上, 在所有的選擇之中), 說一切運作正常. 它似乎包含了內建的印表機支援, 如同一個人所會想要的. Caldera 應該會在 <http://www.caldera.com/> 上提供一些資訊. 你也可以直接向 Corel 所選擇的 Unix 移植公司購買較新版的 WordPerfect.

其它的廠商請別客氣地告訴我你所提供的產品.

15 在螢幕上預覽可列印的東西

幾乎任何你能列印的東西就可以在螢幕上觀看.

15.1 PostScript

Ghosrscript 有一個 X11 的驅動程式最好在 PostScript 的預覽程式 *gv* 的管理下使用. 這些程式的最新版應該也可以看 PDF 檔案. 要注意 gv 已經取代了較舊的預覽程式 "Ghostview"; 新的使用者介面要比 ghostview 平面老舊的 Athena 圖形使用者介面漂亮的多, 功能也比較多.

15.2 TeX dvi

TeX DeVice Independant 檔案可以在 X11 下用 *xdvi* 來預覽. 現代版的 xdvi 會呼叫 ghostscript 來描繪 PostScript 的部份.

有一個 VT100 的驅動程式, 叫做 `dgvt`. 假如你只能做到這樣的話, `Tmview` 可以搭配 Linux 和 svgalib 一起用.

15.3 Adobe PDF

Adobe 的 Acrobat Reader 有提供 Linux 版; 只要從他們的網站 `<http://www.adobe.com/>` 下載就可以了.

你也可以用 xpdf, 它是個免費軟體(freeware) 而且附原始碼, 同時我想 Ghostview 現在應該已經支援在 X11 下用 gs 看 PDF 檔案.

16 在 lpd 下使用序列式印表機

16.1 設定 printcap

Lpd 在 */etc/printcap* 中提供了五個可以設定的屬性, 讓你用來控制印表機所連接的序列埠的設定. 閱讀 printcap 的線上手冊(man page), 並請注意 *br#*, *fc#*, *xc#*, *fs#* 和 *xs#* 的意義. 其中最後的四個屬性是位元組列(bitmap), 用來代表該序列埠的使用設定. 而 *br#* 屬性只是單純的鮑率(baud rate), 例如 ‘`br#9600`’.

要將 *stty* 的設定轉換為 printcap 的旗標設定(flag settings) 非常簡單. 假使你需要, 請現在就看看 stty 的線上手冊.

你可以用 stty 來設定印表機埠, 以便讓你可以將一個檔案 cat 到該埠, 並且使它正確地列印出來. 這兒是我的印表埠 ‘*stty -a*’ 的結果:

```
dina:/usr/users/andy/work/lpd/lpd# stty -a < /dev/ttyS2
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr
-igncr -icrnl ixon -ixoff -iuclc -ixany -imaxbel
-opost -olcuc -ocrnl -onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0
bs0 vt0 ff0
-isig -icanon -iexten -echo -echoe -echok -echonl -noflsh -xcase
-tostop -echoprt -echoctl -echoke
```

這裏和開機時序列埠初始化的設定方式唯一的不同在於 `-clocal`, `-crtscts`, 和 `ixon`. 根據你的印表機的流量控制(flow control) 方式, 你的設定可能會不同.

實際上, 你是以特殊的方式使用 stty. 因為 stty 的對象是連接在標準輸入(standard input) 上的終端機裝置, 你是用如上所示的 ‘$<$’ 字元, 來使得 stty 處理所指定的序列埠.

只要你的 stty 設定正確, '*cat file > /dev/ttyS2*' (在我的例子中) 就可以將檔案送到印表機了, 請看一下這個檔案 /usr/src/linux/include/asm-i386/termbits.h, 它包含了很多的 #define 和一些 struct (你可能想把它 cat 到印表機 (已經可以用了吧, 對不對?), 然後把它當做便條紙). 找到下面這一行:

```
/* c_cflag bit meaning */
#define CBAUD   0000017
```

這一段列出了 *fc#* 和 *fs#* 位元的意義. 你會發現到那兒的名稱 (在鮑率之後) 和 stty 輸出的結果其中一行相符合, 我不是說這會很容易嗎?

找到那些在你的 stty 結果中前面有減號(-) 的設定, 把那些數字(是八進位的) 加起來. 這代表了你要清除的位元, 結果就是你的 *fc#* 能力. 當然你也可以直接用 '`fc#0177777`' (我就是), 因爲你將會在清除之後立刻設定這些位元.

現在對那些在 stty 的結果中沒有減號在前面的設定 (在這一段中所列出的) 做同樣的事. 在我的例子中, 重要的設定有 CS8 (0000060), HUPCL (0002000), 和 CREAD (0000200). 同時也找到代表你的鮑率的旗標 (我的是 0000015), 把它們統統加起來, 在我的例子中會得到 0002275. 這就是你的 *fs#* 能力 ('`fs#02275`' 在我的例子中運作的很好).

用同樣的方式針對這個標題檔(include file) 中的下一段, "c_lflag bits", 做設定和清除的動作. 在我的例子中我不需要設定任何東西, 所以只要用 '`xc#0157777`' 和 '`xs#0`' 就可以了.

16.2 會掉字的舊型序列式印表機

Jon Luckey 指出有些具有 ten-cent 序列介面和小緩衝區的舊型的序列式印表機, 當使用流量控制時, 說停止就是眞的表示停止. 他發現用 *`setserial`* 停用他的 Linux 機器的 16550 序列埠的 FIFO 功能後, 掉字的問題就解決了 (如果你要這樣做, 似乎只要將 uart 的種類指定爲 8250 就可以了).

17 感謝

`smbprint` 的資訊是從 Marcel Roelofs <`marcel@paragon.nl`> 的一篇文章來的.

有關使用 Netware 印表機的 `nprint` 的資訊是由 Michael Smith <`mikes@bioch.ox.ac.uk`> 所提供的.

在 lpd 下使用序列式印表機一節是由 Andrew Tefft <`teffta@engr.dnet.ge.com`> 所提供.

有關 gs 和 gamma 的介紹是 Andreas <`quasi@hub-fue.franken.de`> 送來的.

有關序列埠的 30 秒 closing_waiting 的那兩段是由 Chris Johnson <`cdj@netcom.com`> 所貢獻的.

Robert Hart 送來一些極佳的片段, 是關於設定一部使用 HP 網路印表機的列印伺服器, 我逐字照用.

還要特別謝謝你們之中的很多人, 那些歷年來曾經指出文章中打字, url, 以及種種錯誤的人, 謝謝.