# Linux Printing HOWTO

Grant Taylor <gtaylor+pht@picante.com> Svensk översättning av Marcus Rejås <rejas@swipnet.se>
Version 3.17, 1998/06/17 14:03:27 (översatt 9 juli 1998)

Detta är Linux Printing HOWTO (Linux skriva ut HURDÅ), en samling av information om hur man skapar, förhandsgranskar, skriver ut och faxar något under Linux (och andra Unixliknande system).

## Innehåll

**1 Introduktion** **3**

1.1 Översättares kommentar . . . . . . . . . . 3

1.2 Historia . . . . . . . . . . 3

1.3 Copyright . . . . . . . . . . 4

**2 Hur man skriver ut** **4**

**3 Skrivarrutiner i kärnan** **4**

3.1 lp enheten . . . . . . . . . . 4

3.2 Parport enheten 'parport device' (kärnor >= 2.1.33) . . . . . . . . . . 5

3.3 Seriella enheter . . . . . . . . . . 5

**4 Skrivare som stöds** **6**

4.1 Skrivar-specifika noteringar . . . . . . . . . . 6

**5 Vilken spooling mjukvara?** **7**

**6 Hur det fungerar, grunderna** **7**

**7 Konfigurering, grunderna** **8**

7.1 Traditionell lpd konfiguration . . . . . . . . . . 8

7.2 Filrättigheter . . . . . . . . . . 9

**8 Skaffa utskriftsmjukvara** **9**

**9 Distributioners lösningar** **10**

9.1 Red Hat . . . . . . . . . . 10

9.2 Andra Distributioner . . . . . . . . . . 10

**10 Ghostscript.** **10**

10.1 Anropa Ghostscript . . . . . . . . . . 10

10.2 Ghostscript output justering . . . . . . . . . . 11

10.2.1 Output placering och storlek . . . . . . . . . . 11
10.2.2 Gamma, punktstorlekar, etc. . . . . . . . . . 11

**11 Hur man skriver ut på en skrivare över nätverket 11**

11.1 Till en Unix/lpd värd . . . . . . . . . . 11
11.1.1 Med `lpd` . . . . . . . . . . 11
11.1.2 Med `rlpr` . . . . . . . . . . 12
11.2 Till en Win95, WinNT, LanManager, eller Samba skrivare . . . . . . . . . . 12
11.3 Till en NetWare Printer . . . . . . . . . . 13
11.4 Till en EtherTalk (Apple) skrivare . . . . . . . . . . 13
11.5 Till en HP eller annan ethernet skrivare . . . . . . . . . . 13
11.5.1 Till äldre HPs . . . . . . . . . . 14
11.6 Köra ett *if* för fjärrskrivare . . . . . . . . . . 15
11.7 Från Windows. . . . . . . . . . . 16
11.8 Från en Apple. . . . . . . . . . . 16
11.9 Från Netware. . . . . . . . . . . 16

**12 Windows-only skrivare 16**

12.1 Ghostscripts Windowsbaserade omdirigerare . . . . . . . . . . 17
12.2 Programmet pbm2ppa . . . . . . . . . . 17

**13 Hur skriver man ut till en fax. 17**

13.1 Med ett faxmodem . . . . . . . . . . 17
13.2 Använda Remote Printing Service . . . . . . . . . . 17

**14 Hur generar jag något värt att skriva ut. 18**

14.1 Markeringsspråk (Markup languages) . . . . . . . . . . 18
14.2 WYSIWYG Ordbehandlare . . . . . . . . . . 18

**15 Förhandsgranskning av utskrivbara saker på skärmen. 19**

15.1 PostScript . . . . . . . . . . 19
15.2 TeX dvi . . . . . . . . . . 19
15.3 Adobe PDF . . . . . . . . . . 20

**16 Seriella skrivare under lpd 20**

16.1 Konfigurera i printcap . . . . . . . . . . 20
16.2 Äldre seriella skrivare som tappar tecken . . . . . . . . . . 21

**17 Tack till 21**

# 1 Introduktion

Linux Printing HOWTO borde innehålla allt du behöver veta för att kunna sätta upp utskriftsmöjligheter på din Linuxburk. Som brukligt är, så är det något mer avancerat än Microsoft och Apples 'peka och klicka'-värld, men det är också mer flexibelt och definitivt lättare att administrera för stora LANs.

Detta dokument är strukturerat så att de flesta bara behöver läsa den första halvan (ungefär). Det mesta av det abstrakta och situiationsberoende stoffet kommer i andra halvan av dokumentet och kan lätt lokaliseras i innehållsförteckningen, men förmodligen behöver de flesta läsa kapitel 9 eller 10.

Eftersom version 3.x är en helt omskriven version har mycket av stoffet från tidigare versioner plockats bort. Detta för att den föregående versionen var så stor som 60 utskrivna papper och mycket svårläst. Om du inte hittar svaret på vad du söker här bör du göra på följande sätt: a) sök igenom förgående version av detta dokument på *The Printing HOWTO Home Page* `<http://www.picante.com/~gtaylor/pht/>` och b) skicka mig en rad där det står vad som borde vara med i detta dokument men som inte är det.

*Printing HOWTO Home Page* `<http://www.picante.com/~gtaylor/pht/>` är ett bra ställe för att hitta senaste versionen, den finns också, naturligtvis, från SunSite (`sunsite.unc.edu`) och din lokala spegling av LDP.

## 1.1 Översättares kommentar

Det skall sägas att jag inte är någon expert på att översätta dokument. Men jag gör så gott jag kan. Vissa tekniska ord har mig veterligt inga svenska motsvarigheter, dessa står kvar på engelska. I de fall det är vanligt med ett försvenskat uttal eller stavning av ett ord har jag använt det.

Ett ofta återkommande ord är Daemon. Detta är ett engelskt ord som mig veterligt saknar svensk motsvarighet. Jag använder det engelska ordet med något försvenskat uttal i tal, och det engelska i text. En daemon är ett program som (oftast) startas upp vid systemstarten och sedan körs i bakgrunden för att bevaka och utföra vissa uppgifter.

Jag reserverar mig för översättningsfel, rent språkliga fel och stavfel.

Alla länkar och hänvisningar i Texten går till originaldestinationen, vilket innebär att de leder till engelska dokument.

Överallt i texten utom denna sektion avser ordet 'jag' originalförfattaren.

Hoppas att detta dokument kommer till nytta!

–Marcus Rejås <rejas@swipnet.se>

## 1.2 Historia

Detta är den tredje generationen, vilket betyder att det är tredje gången dokumentet skrivs om. Dokumentets historia kan beskrivas som följer:

1. Jag skrev printing-howto 1992 eftersom det blev för många frågor om utskrifter i comp.os.linux, och postade det i gruppen. Detta kom några månader före HOWTO projektet och var således den första 'mini-FAQen' som kallades 'howto'. Den här utgåvan var i ren ascii.

2. Efter att ha hoppat in i HOWTO projektet blev Printing-HOWTO sammanslagen med en Lpd FAQ av Brian McCauley `<B.A.McCauley@bham.ac.uk>`; vi fortsatte att samarbeta med PHT under ett par år eller så. Vid någon tidpunkt implementerade vi Karl Auer's `<Karl.Auer@anu.edu.au>` arbete. Denna generation av PHT var i TeXinfo, och fanns tillgänglig i PS, HTML, Ascii, och Info.

3. Efter att ha låtiti PHT förfalla i över ett år, och ett misslyckat försök att lämna över det till någon annan gjordes denna omskrivning. Denna generation av dokumentet är i Linuxdoc-SGML (numer känt som Sgml-tools paketet).

## 1.3 Copyright



# 2 Hur man skriver ut

Om du redan har installerat lpd och konfigurerat för att skriva ut på din skrivare, eller om din systemadministratör eller försäljare har gjort det åt dig så behöver du bara lära dig att använda kommandot lpr. *Printing Usage HOWTO* `<http://sunsite.unc.edu/LDP/HOWTO/Printing-Usage-HOWTO.html>` täcker detta, och en del andra kommandom för att manipulera skrivarköer som du troligen vill lära dig.

Men om du har ett nytt system eller ny skrivare så måste du installera och konfigurera skrivar rutiner på ett eller annat sätt innan du kan skriva ut. Läs vidare!

# 3 Skrivarrutiner i kärnan

## 3.1 lp enheten

Linux kärnan (<=2.1.32), förutsatt att du kar kompilerat in eller laddat lp enheten (output av `cat /proc/devices` skall innehålla enheten lp), tillhandahåller en eller flera av */dev/lp0*, */dev/lp1*, och */dev/lp2*. Dessa tilldelas INTE dynamisk utan var och en av dem svarar mot en specifik I/O adress. Detta betyder att din första skrivare kan vara *lp0* eller *lp1* beroende på din hårdvara. Prova båda.

Några få användare har rapporterat att deras dubbelriktade lp portar inte upptäcks om de använder en äldre universalriktad skrivarkabel. Se till att du har en passande kabel.

Man kan inte köra plip och lp drivrutinerna på samma gång på en port. Du kan dock ladda den ena eller den andra när som helst antingen manuellt, eller med kerneld om du har version 2.x (eller senare versioner av 1.3.x) av kärnan. Genom att noggrant ställa in irq och dylikt, kan du antagligen köra plip på ena porten och lp på den andra. Någon gjorde så genom att editera drivrutinen. Jag väntar otåligt på en positiv rapport om någon som lyckats göra detta med bara en smart kommandorad.

Det finns ett litet verktyg som heter `tunelp` med vilket du, som root, kan finjustera Linux lp enheters irq, "polling rate" och andra alternativ.

Om den är inbyggd, i en del 1.3.x och senare kärnor, kan kärnan ta emot `lp=` argumentet för att sätta irq och io adresser:

```
När lp drivrutinen är inbyggd i kärnan kan du använda kommandoraden
till LILO/LOADLIN för att sätta portadresser och irq's som
den drivrutinen skall använda.

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

exempel:   lp=0x378,0    eller   lp=0x278,5,0x378,7 **

Observera att om du använder denna finess måste specificera *alla*
```

```
portar som du vill använda, det finns inga "defaults". Du kan stänga av
en inbyggd drivrutin med lp=0.
```

Om den laddas som en modul i version 2 och sena 1.3.x kärnor är det möjligt att specificera io adresser och irq's på insmod kommandoraden (eller i */etc/conf.modules* för att ha effekt på kerneld) genom att använda den vanliga syntaxen. Parametrarna är `io=port0,port1,port2` och `irq=irq0,irq1,irq2`. Läs manualsidan för *`insmod`* för ytterligare information om detta.

**För dem av er (som jag själv) som aldrig kan hitta standardportnumren när ni behöver dem, de är som i det andra exemplet ovan. Den andra porten (*lp0*) är på 0x3bc. Jag har ingen aning om vilken irq den brukar använda.

Källkoden för Linux parallellportsdrivrutin finns i /usr/src/linux/drivers/char/lp.c.

## 3.2 Parport enheten 'parport device' (kärnor >= 2.1.33)

Med början i kärna 2.1.33 (och tillgänglig som patch för kärna 2.0.30), är lp enheten bara en klient till den nya parport enheten. Fördelen med parportenheten är att den rättar till ett antal problem som den gamla lp enheten led av - den kan dela port med andra drivrutiner, den tilldelar tillgängliga parallellportar dynamiskt till enhetsnummer istället för att kräva en fast kommunikation mellan I/O adresser och portnummer, och så vidare.

Tillkomsten av parport enheten har möjliggjort en hel flock av nya parallell-ports drivrutiner för saker som, Zip drives, Backpack CD-ROMs och diskar, etc. En del av dessa finns också tillgängliga för 2.0.x kärnor; leta runt på nätet.

Jag kommer att täcka parport drivaren mer komplett då jag själv använder en, men tills vidare kan du läsa filen `Documentation/parport.txt` i din källkod till kärnan eller titta på *parport web site* `<http://www.cyberelk.demon.co.uk/parport.html>`.

## 3.3 Seriella enheter

Seriella enheter heter vanligtvis något som */dev/ttyS1* under Linux. Verktyget `stty` låter dig interaktivt studera eller sätta inställningarna för en serieport; `setserial` låter dig kontrollera ytterligare några attribut och konfigurera IRQs och I/O adresser för icke standard portar. Ytterligare behandling av serieportar under Linux hittas i *Serial-HOWTO* `<http://sunsite.unc.edu/mdw/HOWTO/Serial-HOWTO.html>`.

Om du använder en långsam seriell skrivare med flödeskontroll kan du komma att upptäcka att dina utskrifter trunkeras. Detta kan komma av att serieportens normala beteende är att slänga alla oöversända tecken från sin buffer 30 sekunder efter det att portenheten stängts. Bufferten kan hålla upp till 4096 tecken, och om din skrivare använder flödeskontroll och är så långsam att den inte kan ta emot all data ur bufferten inom 30 sekunder efter det att utskriftsmjukvaran stängt porten kommer det sista av din utskrift att gå förlorad. Om kommandot `cat file > /dev/ttyS2` producerar kompletta utskrifter för korta filer och trunkerade för längre kan du ha detta problem.

30 sekundersintervallet kan justeras genom "closing_wait" kommandorads argumentet till setserial (version 2.12 and later). En maskins serieportar initieras vanligtvis genom ett anrop till setserial i startfilen rc.serial. Anropet till skrivarserieporten kan modifieras så att det sätter closing_wait på samma gång som det sätter den portens andra parametrar.

# 4 Skrivare som stöds

Linux kärnan stöder i stort sätt alla skrivare du kan ansluta till en seriell eller parallell port, men det finns saker att se upp med, och skrivare som ej går att använda även om de (elektroniskt sätt) kan kommunicera med Linux. Främst bland dessa inkompatibla skrivare är sådana som förlitar sig på "Windows Printing System". (De brukar vara vagt markerade med "for Windows", eller kallas "GDI"-skrivare.) Dessa skrivare *funkar inte* med Linux. De har inte någon som helst intelligens och förlitar sig på att datorns CPU skall göra mycket av det arbete som traditionellt sköts av skrivarens processor. Olyckligtvis kan dessa uppgifter bara utföras av tillverkarspecifika drivrutiner som bara fungerar under Windows. Så köp inte en sådan för att använda med Linux. (Obsevera, om du redan har en så finns det sätt att få Lunux att skriva till en, men de är ganska krångliga och jag har aldrig provat själv. Se avsnitt 12 i detta dokument för ytterligare behandling av Windows-only skrivare.)

Vad gäller skrivare som *fungerar* under Linux är det bästa en skrivare med PostScript stöd. Nästan all Unix mjukvara som skapar utskrifter, skapar dessa i PostScript format, så uppenbart är det lämpligt att ha en skrivare som stöder PostScript direkt. Olyckligtvis är PostScript stöd ovanligt på andra skrivare än laserskrivare, och är ibland ett kostsamt tillbehör.

Om du inte har den (större) budget som behövs för att skaffa en PostScript skrivare kan du skaffa en skrivare som stöds av Ghostscript, en fri PostScript tolk som används i stället för äkta PostScript stöd i skrivaren. På *Ghostscript Home Page* `<http://www.cs.wisc.edu/~ghost/>` finns en lista över skrivare som stöds och information om nya och experimentella drivrutiner. Bidrag till att förbättra sidan med skrivare som stöder Ghostscript genom att rapportera dina (miss)lyckanden med denna mjukvara.

Adobe har nu ett nytt skrivarspråk som heter "PrintGear" Jag tror att det är ett kraftigt förenklat binärformatspråk med en del arv från PostScript men ingen PostScript kompabilitet. Jag har inte hört något om att Ghostscript skulle stöda det. Men en del av dessa skrivare stöder andra språk som PCL, och dessa fungerar under Linux.

## 4.1 Skrivar-specifika noteringar

Detta avsnitt är, och kommer alltid att vara inkomplett. Men det jag skriver här bör alltid vara korrekt. Så här kommer det:

**Canon BJ-10ex**

En rapport föreslår att Canonskrivaren fungerar mycket bättre i Epson LQ läget (som sätts med en dip-switch). Tydligen blev utskriften avbruten eller felaktig i Canon läge. Å andra sidan, är det mycket tydligt Canonstöd i Ghostscript, så det kan ha varit en lyckoträff.

**Lexmark**

Lexmark (en gång en del av IBM, tror jag) gör många skrivare, många av dem är skrivare enbart för Windows:

- Color Jetprinter 1000
- Color Jetprinter 1020 (non Business Edition)
- Color Jetprinter 2030
- Color Jetprinter 2050
- Color Jetprinter 2070
- Color Jetprinter 7000
- Color Jetprinter 7200

- Winwriter 100
- Winwriter 150c
- Winwriter 200

# 5 Vilken spooling mjukvara?

Tills nyligen var valet för Linuxanvändare enkelt - alla körde samma gamla lpd som nästan ordagrant kom från BSD's Net-2 kod. Även idag har de flesta distributörer med denna mjukvara. Men det börjar på att ändras. SVR4-liknande system inkluderande Sun's Solaris har ett helt annat utskrifts spooling paket, centrerat kring lpsched. Och vissa tecken tyder på att Linuxdistributörer kommer att byta till LPRng, en inte alls lika föråldrad skrivar spooling mjukvara som är tillgänglig fritt. LPRng är mycket lättare att adminitrera för större installationer och har en mindre skrämmande slumpvis kodbas som lpd är full av. Det kan till och med ärligt hävda sig som säkert - det finns inga SUID binärer, och det stöder autenticering via PGP eller Kerberos.

För tillfällert, trots belysningen av nya alternativ, är lpd troligtvis utmärkt för de flesta Linuxanvändare. Trots att det inte är det stiligaste systemet, fungerar det utmärkt då det väl är konfigurerat och det är väl förstått och rikligt dokumenterat i tredje-part böcker om Unix.

Om du vill ha ytterligare information om LPRng, kolla *LPRng Web Page* `<http://www.astart.com/lprng/LPRng.html>`. Framtida versioner av denna HOWTO kommer att inkludera både hur man använder LPRng och vanlig lpd.

# 6 Hur det fungerar, grunderna

För att få utskrivning att fungera bra måste du förstå hur lpd systemet fungerar.

Lpd står för Line Printer Daemon, och refererar i en del texter till både daemonen och hela samlingen av program som hanterar utskriftsspoolingen. Dessa är:

`lpd`

Spooling daemon. En av dessa kör för att kontrollera allt på en maskin, OCH en körs per skrivare medan skrivaren skriver ut.

`lpr`

Användares spooling kommando. Lpr kontaktar lpd och stoppar in ett nytt skrivarjobb i kön

`lpq`

Listar jobben i en skrivarkö.

`lpc`

Lpd system kontroll kommando. Med lpc kan du stoppa, starta, omorganisera, etc skrivarköerna.

`lprm`

`lprm` tar bort et jobb ur skrivarkön.

Så hur sitter det ihop? Jo, när systemet startar upp, startas lpd. Den skannar igenom filen */etc/printcap* för att få reda på vilka skrivare som den skall hantera köer för. Var gång någon kör lpr så kontaktar lpr lpd genom */dev/printer*, och matar lpd med både filen som skall skrivas ut och information om vem som skriver och hur den skall skrivas ut. lpd skriver så ut filen på den passande skrivare som står på tur.

lp systemet var från början konstruerat när de flesta skrivare var radskrivare - dvs, folk skrev mest vanlig ascii. Men det visar sig att bara lite scriptjob behövs för att lpd skall fungera för dagens utskriftsjob, som oftast är i PostScript, eller text, eller dvi, eller...

# 7 Konfigurering, grunderna

## 7.1 Traditionell lpd konfiguration

Den minimala konfigureringen för lpd resulterar i ett system som kan köa filer och skriva ut dem. Det tar ingen hänsyn till om skrivaren förstår dem eller inte, och kommer förmodligen inte att leda till attraktiva utskrifter. Men det är ett första steg mot att förstå hur det funkar, så läs vidare!

För att lägga en skrivarkö till lpd måste du lägga till en post (entry) i */etc/printcap*, och skapa en ny katalog för skrivarkön (spool directory) under */var/spool/lpd*.

En post i */etc/printcap* ser ut så här:

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

Detta definierar en skrivarkö som kallas *lp*, *dj*, eller *deskjet*, köade (spoolade) i katalogen */var/spool/lpd/dj*, utan begränsning på utskriftsjobbens storlek om skriver till enheten */dev/lp0* och som inte lägger ett försättsblad (med namnet på den person som skriver ut, m.m.) till utskriften.

Läs nu manualsidan (man-page) för *printcap*.

Ovanstående ser väldigt enkelt ut, men det finns en hake - såvida inte jag sänder filer som som en desjet 500 förstår kommer denna deskjet att skriva konstiga saker. Till exempel om vi skickar en vanlig Unix text fil till en deskjet får vi en utskrift enligt följande:

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

osv. En PostScript fil som skrivs till denna kö resulterar i en vacker listning av PostScript kommandon, utskrivna med denna trappstegseffekt (staircase effect), men ingen användbar utskrift.

Tydligen behövs något mer, och detta är syftet med filtrering (filtering). De mer observanta av er som läste manualsidan till printcap kanske uppmärksammade spool-attributen *if* och *of*. Just *if*, eller 'input-filter' är precis vad vi behöver här.

Om vi skriver ett litet script som lägger till vagnretur före radbyten och kallar det *filter* kan trappstegseffekten elemineras. Vi måste också lägga in en *if* rad i vår printcap-post här ovan:

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

Ett enkelt filterscript skulle kunna vara:

```
#!perl
# Raden ovan skall ha hela sökvägen till perl
# Scriptet måste vara körbart: chmod 755 filter
while(<STDIN>){chop $_; print "$_\r\n";};
# Eventuellt behöver du avsluta med att mata ut papperet: print "\f";
```

Om vi gjorde som ovan skulle vi få en skrivarkö till vilken vi kunde skriva ut vanliga Unix textfiler och få ett meningsfullt resultat. (Ja, det finns fyra miljoner bättre sätt att skriva detta filter men få som är så illustrativa. Du uppmuntras att göra detta mer effektivt.)

Det enda kvarvarande problemet är det att det inte är alltför 'hett' att skriva ut vanlig text - visst skulle det vara bättre att kunna skriva ut PostScript och annan formaterad eller grafisk typ av utskrift. Jo det är klart att det vore bättre, och det är inte svårt att fixa till. Metoden är helt enkelt ett tillägg till ovanstående radmatnings-fixar filter. Om du kan skriva ett filter som kan acceptera godtyckliga filformat som input och producera DeskJet-kosher output för var och en av dessa har du verkligen ett smart utskrifts system.

Ett sådant filter kallas ett *magic* filter. Bry dig inte om att skriva ett sådant själv såvida du inte skriver mycket underliga saker - Det finns många redan skrivna åt dig på nätet. APS Filter är bland de bästa, eller din Linuxdistribution kan ha ett skrivarkonfigureringsprogram som gör detta väldigt enkelt.

## 7.2 Filrättigheter

På allmän begäran lägger jag in en lista på filrättigheterna på intressanta filer i mitt system. Det finns flera bättre sätt att lösa detta, idealiskt genom att använda bara SGID binärfiler och inte göra allt SUID root, men så här var mitt system från början och det funkar bra för mig. (Uppriktigt sagt, om din leverantör inte ens kan leverera ett fungerande lpd så är du ute på djupt vatten).

```
-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 måste för närvarande köras som root så att den kan ansluta till 'low-numbered lp service port'. Den borde bli UID lp.lp eller något sådant efter anslutning, men jag tror inte att den blir det. Det är inte bra.

# 8 Skaffa utskriftsmjukvara

Många färdigskrivna filterpaket (och annan utskriftsrelaterad mjukvara) finns tillgängliga från *SunSite* `<ftp://sunsite.unc.edu/pub/Linux/system/printing/>`. Program som psutils, a2ps, mpage, dvitodvi, flpr, etc kan alla hittas där.

Ett tag fanns det flera olika mjukvarupaket ute som försökte göra skrivarkonfigureringen enklare. De finns troligen kvar, men en av de bästa och aktuellaste är Andreas Klemm's APS Filter paket, som har menystyrd printcap konfigurerare och hanterar praktiskt taget alla tänkbara typer av indata. Om din distributör inte skickar med ett trevligt konfigureringsprogram, är APS Filter rätt väg.

# 9 Distributioners lösningar

Detta avsnitt är, per definition, inte komplett. Skicka gärna in detaljer om din favoritdistribution.

## 9.1 Red Hat

Red Hat har ett grafiskt verktyg för skrivaradministration (i kontrollpanelen) som kan lägga till nätverksskrivare och lokala skrivare. Det låter dig välja en skrivartyp stödd av ghostscript och en Unix-enhetsfil att skriva till, och installerar en post i */etc/printcap* och skriver ett kort PostScript-och-ascii magic filter baserat på `gs` och `nenscript`. Denna lösning fungerar ganska bra och är enkel att konfigurera för vanligt förekommande fall.

## 9.2 Andra Distributioner

Snälla, skicka mig info om vad andra distributioner gör!

# 10 Ghostscript.

`Ghostscript` är ett otroligt betydelsefullt program för utskrifter under Linux. De flesta unixprogram som genererar utskrifter genererar PostScript, vilket oftast är ett $100 tillbehör till en skrivare. Ghostscript, å andra sidan, är gratis, och genererar din skrivares språk från PostScript. Kombinerat med ditt `lpd` input filter, får du en virtuell PostScript skrivare och förenklar livet avsevärt.

Ghostscript finns i två versioner. Den kommersiella versionen av Ghostscript, kallad Aladdin Ghostscript, får användas fritt för personligt bruk men får inte distribueras med kommersiella Linuxdistributioner. Den är generellt sätt ett år före den fria versionen. Just nu till exempel så stöder den kommersiella Adobe Acrobat's Portable Document Format, medan den äldre inte gör det.

Den fria versionen av Ghostscript är GNU Ghostscript, och är helt enkelt en äldre version av Aladdin ghostscript som vänligt skänkts till GNU. (Heder till Aladdin för detta arrangemang; flera mjukvaruföretag borde stöda free software på detta sätt).

Vad du än gör med `gs`, var riktigt säker på att det körs med argumentet för att omöjliggöra filaccess (`-dSAFER`). PostScript är ett komplett programspråk, och ett dåligt PostScript program skulle kunna ge dig en rejäl huvudvärk.

På tal om PDF, Adobe's Portable Document Format är faktiskt inte mycket mer än organiserad PostScript i en komprimerad fil. Ghostscript kan hantera PDF precis som PostScript. Så du kan bli den första i ditt kvarter med en PDF kapabel skrivare.

## 10.1 Anropa Ghostscript

Vanligtvis körs ghostscript av vilket magic filter du än använder, men i debuggingsyfte är det ofta lämpligt att köra det direkt.

`gs -help` ger dig en kort informativ lista på argument och tillgängliga drivrutiner (observera att listan med drivrutiner är inkompilerad, och inte huvudlistan på alla tillgängliga drivrutiner).

Du kan köra gs i testsyfte som: `gs` *`options`* `-q -dSAFER -sOutputFile=/dev/lp1 test.ps`.

## 10.2 Ghostscript output justering

Det finns flera saker du kan göra om inte gs´s output blir till belåtenhet (faktisk, kan du göra vad f..n som helst eftersom du har källkoden).

### 10.2.1 Output placering och storlek

Placering, storlek och utseende av en bild på en sida kontrolleras av de skrivarspecifika drivrutinen i ghostscript. Om utskriften på dina sidor blir för hoptryckt och kort, eller utdragen och lång, eller för stor med en faktor två, bör du titta i din drivrutins källkod och justera vilka parametrar du än hittar. Olyckligtvis är alla drivrutiner olika, så jag kan inte berätta vad du skall justera, men de flesta är skapligt kommenterade.

### 10.2.2 Gamma, punktstorlekar, etc.

De flesta icke-laser skrivare lider av det faktum att deras punkter är relativt stora. Detta resulterar i att bilder blir för mörka. Om du upplever detta problem skall du använda din egen överföringsfunktion. Skapa följande fil i ghostscripts lib-katalog och lägg dess namn till gs anropet före själva filen. Du måste kanske finjustera värdena för att passa din skrivare. Lägre värden resulterar i ljusare utskrift. Framför allt om din skrivare använder en Floyd-Steinberg algoritm för att rastera färger är troligen lägre värden ( 0.2 - 0.15 ) ett bra val.

```
---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<---
```

Det är också möjligt att fixa skrivare som har något typ av färgfel genom att justera dessa värden. Om du gör något sådant rekommenderar jag att du använder filen colorcir.ps, som finns med ghostscript (i examples/ katalogen), som en provsida..

# 11 Hur man skriver ut på en skrivare över nätverket

En av finesserna med lpd är att den stöder utskrift över nätverket till skrivare fysiskt anslutna till en annan maskin. Med en noggran kombination av filter scripts och tillhörande program kan du använda lpd till att på ett raktframt sätt skriva ut på skrivare anslutna till alla sorters nätverk.

## 11.1 Till en Unix/lpd värd

För att låta andra maskiner skriva till din skrivare måste du lista dem i */etc/hosts.equiv* eller */etc/hosts.lpd*. (Observera att *hosts.equiv* medför andra effekter, var säker på att du vet vad du gör innan du listar någon maskin där). Du kan tillåta bara vissa anvndare på andra maskiner att skriva till din skrivare genom att använda *rs* attributet; läs `lpd` manualsidan för information om detta.

### 11.1.1 Med `lpd`

För att skriva ut till en annan maskin skapar du en */etc/printcap* post som denna:

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

Observera att det fortfarande finns en spoolkatalog på den lokala maskinen som hanteras av `lpd`. Om den andra maskinen är upptagen eller offline kommer utskriftsjobben att vänta i spoolkatalogen tills de kan sändas.

#### 11.1.2 Med `rlpr`

Du kan också använda *rlpr* för att skicka ett utskriftsjob direkt till en kö på en annan maskin utan att behöva gå igenom trånglet med att konfigurera lpd för att fixa det. Detta är användbart framför allt då du skriver till olika skrivare bara någon enstaka gång. Från introduktionen av *rlpr*:

Rlpr använder TCP/IP för att skicka utskriftsjobb till lpd servrar var som helst i ett nätverk.

Till skillnad från lpr kräver rlpr *inte* att den andra skrivaren är känd för maskinen som du vill skriva från (t.ex. genom */etc/printcap*) och kan därför anses vara mer flexibel och kräver mindre administration.

rlpr kan användas överallt där traditionell lpr kan användas och är bakåtkompatibel med traditionell BSD lpr.

Den största styrkan med rlpr är förmågan att skriva ut på fjärrenheter *från var-som-helst till var-som-helst* utan att bry sig om hur det andra systemet du vill skriva ut från är konfigurerat. Kan användas som ett filter precis som traditionell lpr så att klienter som körs på en avlägsen maskin som netscape, xemacs, etc, etc kan skriva ut till din maskin med små ansträngningar.

Rlpr finns tillgängligt från *SunSite* `<ftp://sunsite.unc.edu/pub/Linux/system/printing/>`.

### 11.2 Till en Win95, WinNT, LanManager, eller Samba skrivare

Det finns en Print to Windows mini-HOWTO där ute som innehåller mer info om detta än vad som finns här.

Det är möjligt att leda en lpd kö genom `smbclient` programmet (en del av samba-sviten) till en TCP/IP baserad SMB print service. Samba inkluderar ett script för detta som kallas `smbprint`. Kortfattat går det till så att du lägger en konfigurationsfil för skrivaren i fråga i spool katalogen och installerar `smbprint` scriptet som *if*.

*/etc/printcap* posten ser ut så här:

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

Du skall läsa dokumentationen i `smbprint` scriptet för ytterligare information om hur du konfigurerar detta.

Du kan också använda `smbclient` för att sända en fil direkt till ett SMB printing system utan att blanda in `lpd`. Se manualsidan.

## 11.3 Till en NetWare Printer

ncpfs sviten inkluderar en funktion som kallas `nprint` som tillhandahåller samma funktion som `smbprint` fast for NetWare. Du kan skaffa ncpfs från `SunSite`. Utdrag ur LSM posten för version 0.16:

> 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.

För att få `nprint` att fungera via lpd måste du skriva ett litet script som skriver stdin på NetWare skrivaren och installera detta som *if* till en lpd skrivarkö. Du får något som liknar:

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

`nprint-scriptet` borde se ut ungefär så här:

```
#! /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 Till en EtherTalk (Apple) skrivare

Netatalk paketet innehåller någonting som liknar `nprint` och `smbclient`. Andra har dokumenterat proceduren för att skriva till och från ett Apple nätverk mycket bättre än vad jag någonsin kommer att göra, se *Linux Netatalk-HOWTO* `<http://thehamptons.com/anders/netatalk/>`.

## 11.5 Till en HP eller annan ethernet skrivare

HP och några andra skrivare levereras med ett ethernet interface som du kan skriva till direkt med lpd. Du skall följa instruktionerna som följer skrivaren eller dess nätverskadapter. Generellt sett ”kör” dessa skrivare lpd och har en eller flera köer till vilka du kan skriva. En HP, till exempel, borde fungera med en printcap post som:

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

HP Laserjet skrivare med Jet Direct interfaces har vanligtvis två stycken inbyggda lpd köer - ”raw” som accepterar PCL (och möjligtvis PostScript) och ”text” som accepterar ren ascii (och hanterar automatiskt trappstegseffekten). Om du har en JetDirect Plus3 tre-portsburk, heter köerna ”raw1”, ”text2”, o.s.v.

I en storskalig miljö, framför allt där inte alla skrivare stöder PostScript, kan det vara lämpligt att utse en speciell skrivarserver till vilken alla maskiner skriver och på vilken alla ghostscript jobb körs.

Detta låter också din Linuxburk att agera som en spool server till skrivaren så att dina nätverksanvändare snabbt kan avsluta sina utskriftsjobb och forsätta med annat utan att vänta på att skrivaren skall skriva ut något som någon annan har skickat.

För att åstadkomma detta, sätt upp en kö på din Linuxburk som pekar på den ethernetutrustade HP LJ (som ovan). Sen konfigurerar du alla klienter på ditt LAN att skriva till denna kö (ex. lj-5 i exemplet ovan).

Till en del HP nätverksskrivare behöver troligen inte försättsbladsinställningen skickas av klienterna, du kan stänga av deras internt genererade försättsblad genom att 'telnetta' till printern, trycka retur två gånger, skriva "banner: 0" följt av "quit". Det finns flera inställningar som du kan ändra på detta vis, skriv "?" för att se en lista.

### 11.5.1 Till äldre HPs

En del skrivare (och skrivarrutiner i nätverk som "black boxes") stöder bara ett dåligt litet icke-protokoll med rena TCP anslutningar. Att märka i denna kategori är tidiga JetDirect (inkluderande en del JetDirectEx) kort. Kortfattat, för att skriva till skrivaren måste du öppna en TCP anslutning till skrivaren på en speciell port (vanligtvis 9100 eller 9100, 9101, 9102 för tre-ports) och stoppa in ditt utskriftsjobb i skrivaren. Detta kan, bland annat, implementeras med Perl:

```
#!/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 Köra ett *if* för fjärrskrivare

En underlighet men lpd är att *if* inte körs för fjärranslutna skrivare. Om du finner att du behöver köra ett *if*, så kan du göra detta genom att konfigurera en dubbel kö och köa om jobbet. Som ett exempel studera följande *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:
```

belyst av detta `filter-lj-5` script:

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

Argumentet *-U* till lpr fungerar bara om lpr körs som en daemon och sätter begärarens namn för jobbet i den andra kön på rätt sätt. Du bör använda en mer rubust metod för att få användarnamnet eftersom det i vissa fall inte är argument 5. Se manualsidan för ***printcap***.

### 11.7 Från Windows.

Att skriva från en Windows (eller förmodligen OS/2) klient till en Linuxserver är direkt stödd över SMB genom användande av SAMBA paketet, som dessutom stöder fildening av ditt Linux filsystem till Windowsklienter.

Samba inkluderar en ganska komplett dokumentation, och det finns en bra Samba FAQ som också täcker det. Du kan antingen konfigurera ett magic filter på Linuxburken och skriva PostScript till detta eller springa runt och installera skrivarspecifica drivrutiner på alla Windowmaskiner och ha en kö för dem utan något filter. Att förlita sig på Windows drivrutiner kan i en del fall leda till bättre utskrifter, men ar mer administrativt krävande om det är många Windowsmaskiner. Du bör testa PostScript först.

### 11.8 Från en Apple.

Netatalk stöder utskrift från Apple klienter över EtherTalk. Se *Netatalk HOWTO Page* `<http://thehamptons.com/anders/netatalk/>` för mer information.

### 11.9 Från Netware.

ncpfs paketet inkluderar en daemon som heter pserver som kan användas till att serva en NetWare skrivarkö. Vad jag har förstått kräver ett sådant system "Bindery-Based NetWare" d.v.s. 2.x, 3.x eller 4,x med "bindery access" aktiverad.

För ytterligare information om ncpfs och dess pserver progam, se *the ncpfs FTP site* `<ftp://ftp.gwdg.de/pub/linux/misc/ncpfs/>`.

## 12 Windows-only skrivare

Som jag tidigare nämnt är en del skrivare i sig inte stödda i Linux eftersom de inte talar normalt skrivarspråk utan istället använder datorns CPU för att skapa bitmappar som sedan skickas till skrivare i en bestämd hastighet. I några få fall talar dessa skrivare också något normalt, som PCL, men oftast inte. I en del (verkligt lågklass) fall använder inte skrivaren ens en normal parallellanslutning utan förlitar sig på att tillverkarens drivrutin skall emulera hur hårdvaran skall uppföra sig (det viktigaste är handskakningen 'handshaking').

I alla fall, det finns några få trix att komma runt problemet om du har fastnat med en sådan otrevlighet.

### 12.1 Ghostscripts Windowsbaserade omdirigerare

Det finns nu en Windows skrivardrivrutin tillgänglig (heter mswinpr2) som kör ett skrivarjobb genom Ghostscript innan det slutligen skriver ut det. (Istället för ett if filter i Unix's LPD). Det finns också en ny Ghostscript drivrutin som skriver genom att använda Windows GDI anrop. Tillsammans, bör detta tillåta en Windows maskin att skriva PostScript till en Windows-only skrivare genom tillverkarens drivrutin.

Om du får det att funka, kan du följa instruktionerna ovan för att skriva till en Windows skrivare över nätverket från Linux för att låta Unix (och andra Windows, Mac, etc) värdar att skriva till din otrevliga skrivare.

Det skall sägas att jag aldrig testat detta eftersom jag inte har någon Windowsskrivare, så du får testa själv. All inblandad mjukvara skall gå att hitta på Ghostscript's hemsida (se länkarna i Ghostscriptavsnittet i detta dokument).

### 12.2 Programmet pbm2ppa

En del HP skrivare använder "Printing Performance Architecture" (marknadsföringsspråk för "vi är för billiga för att implementera PCL"). Detta stöds via en omväg över pbm2ppa översättaren, skriven av Tim Norman. Vanligtvis använder du Ghostscript för att översätta PostScript till en bitmappad bild i pbm format och sedan använder pbm2ppa för att översätta denna till ett skrivarspecifikt ppa format klart för att dumpas till skrivaren. Detta program kan också ha kommit i en Ghostscript drivare vid det här laget.

ppa mjukvaran kan hämtas från *the ppa home page* `<http://www.rpi.edu/~normat/technical/ppa/>`. pbm2ppa stöder en del modellar av HP 720, 820, och 1000; läs dokumentationen som följer med paketet för detaljer om ppa skrivarstöd.

## 13 Hur skriver man ut till en fax.

### 13.1 Med ett faxmodem

Det finns ett flertal faxprogram där ute som låter dig faxa och ta emot dokument. Ett av de mest komplexa är Sam Leffler's *HylaFax*, tillgängligt från `ftp.sgi.com`. Det stöder allt från flera modem till broadcasting (=sända till flera samtidigt).

SuSE levererar en Java HylaFax klient som påstås fungera på vilken Java plattform som helst (inkluderande Windows och Linux). Det finns också icke-Java fax klienter till de flesta plattformar, Linux kan med största sannolikhet hantera dina nätverks fax behov.

Dessutom finns, och ett bättre val för de flesta Linuxburkar, `efax`, ett enkelt program som skickar fax. Getty-programmet `mgetty` kan ta emot fax (och till och med voicemail på en del modem!).

### 13.2 Använda Remote Printing Service

Det finns en experimentell service som låter dig sända ett e-postmeddelande som innehåller det du vill skriva ut på en fax någonstans. Trevliga format som postscript stöds, så även om global täckning är gles kan detta vara en användbar service. För ytterligare information om att skriva ut till en fax via remote printing service, se *Remote Printing WWW Site* `<http://www.tpc.int/>`.

# 14 Hur generar jag något värt att skriva ut.

Här går vi in i ett riktigt råttbo av mjukvara. Normalt kan Linux köra flera typer av binärfiler med varierande grad av framgång: Linux/x86, Linux/Alpha, Linux/Sparc, Linux/foo, iBCS, Win16/Win32s (med dosemu och, någon dag, med Wine), Mac/68k (med Executor), och Java. Jag kommer bara att ta upp rena Linux och vanliga Unix program.

Till Linux självt faller valet oftast begränsat till de program som normalt finns till Unix:

## 14.1 Markeringsspråk (Markup languages)

De flesta markeringsspråk passar bäst till stora eller repetitiva projekt där du vill att datorn skall ta hand om layouten av texten för att göra den enhetlig. Att försöka göra ett snyggt tecken i ett markeringspråk kommer förmodligen att göra ont...

`nroff`

Detta var ett av de första markeringsspråken till Unix. Manualsidorna är de vanligaste exemplen på jobb som formaterats med *roff macros; många älskar dem, men nroff har, åtminstone för mig, en krångligare syntax än nödvändigt, och är förmodligen ett dåligt val för nya arbeten. Det är dock värt att veta att du kan typsätta en manualsida direkt till postscript med groff. De flesta man kommandon gör det åt dig med `man -t foo | lpr`.

**TeX**

TeX, och macropaketet LaTeX, är en av de mest använda markeringsspråken använda i Unix. Tekniska arbeten skrivs ofta i LaTeX eftersom det verkligen underlättar layoutproblem och är *forfarande* ett av de få ordbehandlar system som stöder matematik både komplett och bra. TeX's output format är `dvi`, som konvereras till PostScript eller Hewlett Packard's PCL med `dvips` eller `dvilj`.

**SGML**

Det finns åtminstone en fri sgml analyserare för Unix och Linux; den formar grunden för Linuxdoc-SGML's egna dokumentsystem. Det kan stödja andra DTD´s (dokument-typ-deklarationer) också.

**HTML**

Någon föreslog att man för enkla projekt kan skriva i HTML och sedan skriva ut med Netscape. Jag håller inte med, men det står för mig.

## 14.2 WYSIWYG Ordbehandlare

Det är inte längre någon brist på WYSIWYG ordbehandlings mjukvara. Flera kompletta Office paket finns tillgängliga. Ett av dem är gratis för personligt bruk (StarOffice).

**StarOffice**

Ett tyskt företag distribuerar StarOffice på Internet gratis för Linux. Detta fullfjädrade office paket har allt du kan förvänta dig, och du slår inte priset. Det finns en mini-HOWTO där ute som beskriver hur du skaffar och installerar det. Det generear PostScript eller PCL så det fungerar med i stort sätt alla skrivare som normalt fungerar med Linux.

**LyX**

LyX är ett skal till LaTeX som verkar mycket lovande. Se the *LyX Homepage* `<http://www.lyx.org/>` för mer information. Det finns en KDE version av LyX som heter KLyX. Författaren till Lyx och upphovsmannen till KDE är samma person.

**The Andrew User Interface System**

AUIS inkluderar `ez`, en WYSIWYG-liknande editor med de vanligaste ordbehandlings tillbehören, HTML kapabilitet coh full MIME e-post och newsgroup stöd. Tyvärr underhålls inte AUIS längre.

**Koffice**

The KDE project arbetar på en hel office svit. Jag tror inte att den är redo för att presenteras ännu.

**GNOME**

The GNOME project arbetar också på några GNU-licensierade office-verktyg. Inget finns tillgängligt ännu, dock.

**Kommersiella alternativ**

Åtminstone Caldera och Red Hat säljer programpaket innehållande de vanliga office applikationerna som en WYSIWYG ordbehandlare och ett kalkylprogram. Jag antar att de gör ett snyggt jobb, men jag har aldrig testat dem. Jag tror att Caldera säljer Sun's WABI också, så du kan förmodligen köra någonting som MS Office under det om du måste integrera med andras filer.

Jeff Phillips <`jeff@I_RATUS.org`> använder Caldera's WordPerfect för Linux (på Slackware, av alla) och säger att det fungerar bra. Det inkluderar tydligen inbyggt skrivarstöd, som man borde förvänta sig. Caldera borde ha mer information på `<http://www.caldera.com/>`. Du kan också köpa en nyare version av WordPerfect direkt från en av Corels utvalda Unix återförsäljare.

RedHat och SuSE levererar ett paket som heter *Applixware*; du hittar deras hemsida på `<http://www.redhat.com/>`. Applix finns tillgänglig till många plattformar och inkluderar, kalkylprogram, ordbehandlingsprogram, presentationspaket, osv. Det finns en separat version av Applix tillgänglig i Java som heter ApplixAnyWhere, jag antar att den är ganska lika som den andra men långsammare och med mindre finesser.

Andra handlare kan gärna skicka mig en rad med deras erbjudanden.

# 15 Förhandsgranskning av utskrivbara saker på skärmen.

Nästan allt du kan skriva ut kan också förhandsgranskas på skärmen.

## 15.1 PostScript

Ghostscript har en X11 drivrutin som fungerar bäst under kontroll av PostScript förhandsgranskaren `Ghostview`. De senaste versionerna av dessa program borde kunna visa PDF filer också. Märk att gv har bytt ut den gamla förhandsgranskaren "Ghostview", det nya användargränssnittet är mycket snyggare och har mer finesser än det enkla gamla Athena gränssnittet.

## 15.2 TeX dvi

TeX DeVice Independant filer kan förhandsgranskas under X11 med `xdvi`. Moderna versioner av xdvi anropar ghostscript för att rendera PostScript specialare.

En VT100 drivrutin finns också. Den heter `dgvt`. `Tmview` fungerar med Linux och svgalib, om det är allt du kan åstadkomma.

### 15.3 Adobe PDF

Adobe's Acrobat Reader finns till Linux; ladda bara hem den från deras hemsida `<http://www.adobe.com/>`.

Du kan även använda xpdf, som är freeware och levereas med källkod, och jag tror att Ghostview stöder visning av PDF filer med gs under X11 vid den här tiden.

## 16 Seriella skrivare under lpd

### 16.1 Konfigurera i printcap

Lpd tillhandahåller fem attribut vilka du kan sätta i */etc/printcap* för att styra inställningarna av den seriella port som skrivaren sitter på. Läs ***printcap*** manualsidan och notera betydelsen av *br#*, *fc#*, *xc#*, *fs#* och *xs#*. De siste fyra av dessa attribut är bitmaps som indikerar inställningarna för att använda porten. Till exempel *br#* atrributet är helt enkelt baudtalet t.ex. '`br#9600`'.

Det är väldigt enkelt att översätta från `stty` inställningar till printcap argumentinställningar. Om du behöver, titta i manualsidan för stty nu.

Använd stty för att konfigurera skrivarporten så att du kan 'cat' en fil till den och den skrivs ut korrekt. Då här ser '*stty -a*' ut för min skrivarport:

```
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
```

De enda skillnaderna mellan denna och det sätt porten initieras vid systemstarten är `-clocal`, `-crtscts`, och `ixon`. Din port kan vara annorlunda beroende på hur din skrivare gör flödeskontroll.

Du använder faktiskt stty på ett på någotvis udda sätt. Eftersom stty kör mot terminalen som är kopplad till dess standard input, använder du det till manipulera en given serieport genom att använda tecknet '<' enligt ovan.

När du väl fått stty inställningarna rätt, så att '*cat fil > /dev/ttyS2*' (i mitt fall) sänder filen till skrivaren, tittar du i filen /usr/src/linux/include/asm-i386/termbits.h. Denna innehåller en en hel del #defines och några structs. (Du kanske vill 'cat'a denna fil till skrivaren (det funkar ju nu, eller hur?) och använda den som ett kladdpapper) Gå till sektionen som börjar med

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

Denna sektion listar betydelsen av *fc#* och *fs#* bitarna. Du kommer att märka att namnen där (efter baudtalen) matchar en av raderna av stty output. Sa jag inte att detta skulle bli enkelt?

Notera vilken av inställningarna som inleds med ett - i din stty output. Summera ihop alla dessa siffror (de är oktala). Detta representerar de bitar som du vill "clear"a, så resultatet är din *fc#* möjlighet. Naturligvis, kom ihåg att du kommer att sätta dessa bitar direkt efter du har "clear"at, så du kan använda '`fc#0177777`' (Det gör jag).

Nu gör du detsamma för inställningarna (listade i denna sektion) som inte har ett - före dem i din stty output. I mitt exempel är de viktiga CS8 (0000060), HUPCL (0002000), och CREAD (0000200). Notera också argumenten för ditt baudtal (mitt är 0000015). Lägg ihop dessa, och i mitt exempel får du 0002275. Det skall in i din *fs#* möjlighet ('`fs#02275`' funkar bra i mitt exempel).

Gör samma sak med set och clear för nästa sektion av include filen, "c_lflag bits". I mitt fall behövde jag inte "set"a något så jag använde bara '`xc#0157777`' och '`xs#0`'.

### 16.2 Äldre seriella skrivare som tappar tecken

Jon Luckey pekar ut att några äldre seriella skrivare med billiga seriella interface och små buffers *verkligen* menar stopp när de säger så med flödeskontrollen. Han kom på att stänga av FIFO'n på hans Linuxburks 16550 serieportar med `setserial` fixade problemet med tappade tecken (du till synes bara specificerar uart typen till 8250 för åstadkomma detta).

## 17 Tack till

`smbprint` informationen är från en artikel av Marcel Roelofs <`marcel@paragon.nl`>.

`nprint` informationen för att använda Netware skrivare tillhandahölls av Michael Smith <`mikes@bioch.ox.ac.uk`>.

Seriella skrivare under lpd sektionen är från Andrew Tefft <`teffta@engr.dnet.ge.com`>.

Stoffet om gammas och sådant för gs skickades in av Andreas <`quasi@hub-fue.franken.de`>.

De två paragraferna om 30 sekunders closing_wait för den seriella drivrutinen tillhandahölls av Cris Johnson <`cdj@netcom.com`>.

Robert Hart skickade in några utmärkta paragrafer om konfigureringen av en utskriftsserver till nätverks HPs som jag använde ordagrant.

Och speciellt tack till er dussin på dussin som påpekat stavfel, brustna URL:er och fel i dokumentet över åren.