Impresión Como

Grant Taylor gtaylor@cs.tufts.edu y Brian McCauley (B.A.McCauley@bham.ac.uk
Traducción: Francisco Escarpa escarpa@impronta.es

Una guía de cómo imprimir y ver documentos bajo el sistema operativo Linux.

Índice General

1 Introducción

Esta es la versión española del Printing-HOWTO de Linux. Éste documento forma parte de la segunda generación de linux FAQ[1] El *Linux-FAQ* original devino en una bestia inmanejable, por lo que se cambió el esquema y se fraccionó en secciones independientes para cada aspecto del sistema operativo.

Este *COMO* en particular trata de cómo administrar los distintos aspectos de impresión bajo Linux, configurar impresoras, instalar aplicaciones para hacerlas funcionar... Originalmente escrito por Grant Taylor `gtaylor@cs.tufts.edu`), posteriormente incorporó el *lpd-FAQ* de Brian McCauley (`B.A.McCauley@bham.ac.uk`).

Otros *COMO*s detallan aspectos acerca de la conexión en red.... Aunque pueden encontrarse estos documentos en los lugares más insospechados, su "domicilio" oficial es:

`ftp://sunsite.unc.edu/pub/linux/docs/Howto`[2]

(busque mejor en los Mirrors. Sunsite es una playa muy concurrida) Junto con el resto de los FAQ de usenet, podemos encontrarlos en `ftp://rtfm.mit.edu`

Envíen los comentarios, errores detectados y sugerencias a Brian y Grant a `printing@god.ext.tufts.edu` (versión en inglés) o a Francisco Escarpa para la traducción española (`escarpa@impronta.es`).

Les rogamos que nos comuniquen cualquier nuevo programa, fichero, ... que conozcan y no figure en este documento.

Grant ha configurado un servidor de correo centrado en la impresión y visualización bajo Linux, incluyendo la última versión de este documento (en inglés). Para acceder, envíe un mensaje de correo electrónico a `listserv@god.ext.tufts.edu`, con el cuerpo de mensaje '`INFO`'. Recibirá como respuesta una lista de los ficheros disponibles. Si el cuerpo del mensaje es '`get fich1 [fich2 fich3]`', recibirá una copia de los ficheros pedidos.

Para esta traducción se ha reelaborado ligeramente el documento original, eliminado repeticiones, reducido los detalles particulares de cada programa al mínimo imprescindible, remitiendo al lector a la documentación correspondiente, y añadiendo alguna pequeña novedad conocida. Agradeceremos las sugerencias, notificación de errores y demás.

2 Prepárense los impacientes.

Usted ya conoce UNIX, y ya tiene configurado el sistema (o se lo han configurado) y desea con toda urgencia imprimir: Hágalo con la orden:

```
lpr fichero
```

y por su impresora saldrán las tan ansiadas páginas. Si no funciona, lea de cabo a rabo este documento.

3 Puertos y dispositivos

Como es sabido, Unix mapea los dispositivos sobre el sistema de ficheros; así, cada puerto físico está reflejado en un fichero, generalmente bajo `/dev`, sobre el que se ejecutan las operaciones de e/s.

En un sistema de bus tipo AT, común a todos los PC actuales, el puerto *LPT1* equivale en Linux al fichero de dispositivo `/dev/lp1` y sucesivos.

[1] Nota del Revisor
O *PUF*s, *Preguntas de Uso Frecuente*

[2] Igualmente, la mayoría de los *mirrors* de Sunsite contendrán las traducciones de los COMO, en su directorio `ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/translations/es`

Precisando:

Nombre	Mayor	Menor	Direccion E/S
lp0	6	0	0x3bc
lp1	6	1	0x378
lp2	6	2	0x278

Del mismo modo, los puertos *COM* se controlan a través de los dispositivos `ttyS*` (Consulte el Serie-Como, en `http://www.insflug.org` para más información)

Utilice los dispositivos `/dev/ttyS?` (o `/dev/ttys?` si lo prefiere) para conectar una impresora por línea serie. **No utilice** `/dev/cua?`.

En algunos sistemas, puede haber enlaces simbólicos hacia estos dispositivos, como `/dev/lpr` -> `/dev/lp1`, o `/dev/impserie` -> `/dev/ttyS0`. Hay a quien le parece muy elegante, y quien lo considera de pésima praxis. (lea la página `man` *'ln'* para más información).

Recuerde que NO puede simultáneamente tener soporte para impresora y para comunicaciones vía puerto paralelo (conectarse con `plip`, o un disco ZIP paralelo, por ejemplo); si necesita ambas funcionalidades es una buena idea recompilar el kernel, y preparar el soporte de impresión y de comunicaciones como módulos independientes, que se cargarán en el kernel cuando sea necesario.

4 LPR

Todo sistema que se precie es capaz de gestionar una o varias impresoras, con uno o varios usuarios, que les envían distintas clases de documentos, más o menos dignamente.

Unix resuelve estos problemas mediante un conjunto de programas, los servidores de impresión, que gestionan los trabajos pendientes, y los encauzan a las impresoras adecuadas, todo de manera completamente transparente al usuario.

Hay dos familias de servicios de impresión: "`lp`" en UNIX System V (AT&T), y "`lpr`", en los U*X BSD, y Linux. En este documento sólo se hablará del segundo sistema; el sistema `lp` no es de dominio publico, y no conocemos ninguna versión que lo sea; de todos modos, ambos tienen hoy día la misma funcionalidad.

Una variante muy prometedora de `lpr`, *LPRng*, obvia toda una serie de limitaciones del `lpr` original. Nos limitaremos a mencionar las diferencias en los capítulos concretos.

Dicen los que saben que `lp` es más robusto, pero a la hora de trabajar en red `lpr` se destaca por varios cuerpos.

4.1 ¿Qué necesita?

Asumimos que usted sabe cómo editar un archivo de texto bajo Linux, y tiene los conocimientos básicos sobre permisos y propiedad de archivos.

También suponemos que su sistema Linux funciona sin tropiezos. Si desea imprimir en máquinas remotas, deberá además tener configurado y funcionando el soporte de red (vea en `NET-3-HOWTO`[3]).

Revise las páginas de manual de los comandos `chmod` y `chown` para más información.

[3]N. del Revisor:
Disponible en breve en castellano como `NET-3-Como`, ver sección 8

4.2 ¿Cómo imprimir con Linux?

El camino más corto en UNIX (y bajo Linux), es enviar los datos a imprimir directamente al dispositivo adecuado. El siguiente comando envía un listado del directorio a la primera impresora en paralelo (hablando en DOS, LPT1:):

```
ls > /dev/lp1
```

El problema de este método es que no aprovecha las capacidades de multitarea de Linux, debido a que el tiempo que tarda el comando en completarse será el mismo que emplee la impresora en despachar el trabajo.

En una impresora lenta, o en una apagada o sin papel, puede prolongarse un poco. Podríamos ejecutar el comando simplemente en segundo plano, pero no adelantaríamos mucho. Además, deberá tener privilegios de superusuario.

Un remedio mejor es crear un área tampón (*spool*), es decir, guardar los datos a imprimir en un fichero temporal, y arrancar un proceso en segundo plano que envíe los datos a la impresora, y gestione las incidencias que se presenten.

Esencialmente, así funciona Linux. Para cada impresora, se define un área tampón, donde cada trabajo pendiente se almacena en un fichero. Un proceso en segundo plano (llamado el demonio de impresión) analiza metódica y constantemente los ficheros tampón, buscando nuevos datos a imprimir. Cuando aparece alguno, son enviados a la impresora apropiada; cuando más de un fichero está a la espera, se colocan en una cola (el primero que entra es el primero que se procesa), por lo que se habla propiamente de la "*cola de impresión*".

En el caso de impresión remota, los trabajos se gestionan localmente, como cualquier otro, pero el demonio de impresión lo envía a través de la red hacia el ordenador o impresora destino.

La información que el demonio de impresión necesita para su trabajo (el dispositivo físico, el tampón de datos, la máquina e impresora remota ...) se almacenan en un fichero llamado "`printcap`", que describiremos más tarde.

En lo sucesivo, el término "impresora" se referirá a una máquina lógica definida en `/etc/printcap`. El concepto "impresora física" o "trasto", define la cosa que mancha papel. Es perfectamente posible describir múltiples entradas en `/etc/printcap` que se refieren a un sólo trasto, pero por caminos tortuosos. No se preocupe, lo aclararemos al describir `printcap`.

4.3 Impresión remota e impresión local.

La impresión remota nos permite enviar trabajos de impresión desde una máquina, hacia otra (ordenador/impresora) conectada a una red; por ejemplo, nuestro equipo funciona de servidor en una red, o si una impresora asignada a nuestra máquina debe ser accesible por otros ordenadores.

Imprimimos localmente cuando usuarios de nuestra máquina envían trabajos a una impresora conectada directamente a la misma.

4.4 Los programas importantes

El sistema de impresión de Linux se sustenta en cinco programas, que deberían estar donde aparecen en el siguiente listado, propiedad de `root`, y grupo `lp` (o `daemon`, según el sistema en concreto):

```
-rwxr-xr-x    root    lp    /bin/lpr

-rwxr-xr-x    root    lp    /bin/lpq
```

```
-rwxr-xr-x    root    lp    /bin/lpc

-rwxr-xr-x    root    lp    /bin/lprm

-rwxr-x---    root    lp    /sbin/lpd
```

Los cuatro primeros tienen por fin enviar, cancelar y examinar los trabajos de impresión. `/sbin/lpd` es el demonio de impresión.

OJO: Los directorios, permisos y propiedad de los ficheros pueden diferir a los de su sistema, aunque no deberían ser MUY distintos.

Hay páginas de manual que explican con detalle todas estas órdenes y que debería consultar para ampliar información.

Es importante saber que, por defecto, `lpr`, `lprm`, `lpc` y `lpq` trabajan con una impresora llamada "`lp`". Si define la variable de entorno `PRINTER` con el nombre de una impresora, pasará a ocupar el valor por defecto. Se puede indicar sobre la marcha una impresora distinta con la opción `-P impresora` en la línea de órdenes.

4.4.1 La orden `lpr`

Con `lpr` se envía un trabajo a la impresora. Este se copia en el tampón, donde el demonio de impresión lo encuentra, y lo envía a la impresora física. Si no le suministra un fichero, `lpr` usará la entrada estándar.

4.4.2 La orden `lpq`

`lpq` muestra los trabajos pendientes para la impresora deseada ("`lp`" por defecto). `lpq` muestra el número de cada trabajo, que lo identifica para cualquier proceso posterior.

Muestra también el estado de cada trabajo. "`active`" indica que el demonio está enviando el trabajo a su destino, o al menos lo intenta. Si no, un número indica su orden en la cola de impresión.

4.4.3 La orden `lprm`

`lprm` elimina un trabajo de la cola, es decir, borra los ficheros en espera en el directorio tampón. Puede indicar específicamente la identidad de un trabajo particular, o "`-`", con lo que cancelamos todos los trabajos destinados a la impresora seleccionada .Si se es superusuario, y quiere eliminar todos los trabajos pertenecientes a un usuario, especifique su nombre de usuario en la línea de órdenes.

4.4.4 la orden `lpc`

Con `lpc` podemos comprobar el estado de las impresoras, y controlar algunos aspectos de su uso. Particularmente, le permite arrancar y parar el tampón de datos, permite activar y desactivar impresoras, y reorganizar el orden de los trabajos en cola. Con las órdenes siguientes desactivamos la impresora "`impre`", activamos la cola de "`tuimpre`", y mueve el trabajo 37 al principio de la cola.

```
lpc down impre

lpc enable tuimpre

lpc topq 37
```

Si no especificamos argumentos, `lpc` entrará en modo diálogo. con "?" obtenemos ayuda. Advierta que algunas funciones de `lpc` están reservadas para el superusuario.

4.5 Los directorios fundamentales.

Realmente, sólo hay un directorio importante: el área de tampón donde se almacenan los datos a la espera de que `lpd` decida qué hacer con ellos. Sin embargo, un sistema típico debería configurarse en varios directorios, uno para cada impresora, lo que facilita notablemente el mantenimiento. En la mayoría de las instalaciones, `/var/spool/lpd` es el directorio tampón principal, y cada impresora tiene un subdirectorio particular, con el mismo nombre que la impresora. Así, si tiene una impresora llamada PJL-16 que admite PostScript y HPGL, debería crear dos directorios, por ejemplo `/var/spool/lpd/PJL-16-ps` y `/var/spool/lpd/PJL-16-hpgl`.

Los directorios temporales deberían pertenecer a `root`, grupo `lp`; user y group deben poder leer y escribir, y el resto, leer. permisos:

```
-rwxrwxr-x
```

(775)

Para cada directorio de impresora, la orden adecuada sería:

```
chmod ug=rwx,o=rx PJL-16-ps
chgrp lp myprinter
```

Los destinos, permisos y propietarios aquí indicados deben considerarse como indicativos, pues pueden variar entre distintos sistemas e instalaciones.

4.5.1 Y sus correspondientes ficheros.

Además de los programas ya tratados, cada directorio temporal debe contener como mínimo estos cuatro ficheros: "`.seq`" , "`errs`" , "`lock`" y "`status`". Deberán tener permisos

```
-rw-rw-r--
```

(664).

El fichero `.seq` contiene la secuencia de trabajos enviados. `status` contiene el mensaje que devuelve "`lpc stat`". El fichero `lock` impide al `lpd` imprimir al tiempo dos trabajos en la misma impresora, y `errs` guarda un registro de los fallos de la impresora.

El fichero "`errs`" es actualmente potestativo, y ahora puede llamarse como le apetezca su nombre se especificará en `/etc/printcap`. Debe, sin embargo, existir un fichero que permita a `lpd` registrar los mensajes de error.

4.6 `/etc/printcap`

`/etc/printcap` es un fichero texto, modificable con su editor favorito.

Su propietario debe ser `root` y debe tener permisos

```
-rw-r--r--
```

(644)

Aunque a golpe de vista parezca tan comprensible como la piedra Rosetta , su estructura es muy sencilla y asequible. Parte de la mala fama se debe a que algunas distribuciones no incluyen página de manual para `printcap`, y el hecho de que muchos `printcap` están generados por programas, o por gente cuya manera de despreciar al género humano es omitir comentarios que ayuden a su compresión. Desde aquí hacemos un llamamiento para que su fichero `printcap` sea tan legible como sea posible.

Cada entrada de `printcap` describe una impresora. Mejor aún, cada entrada de `printcap` provee una denominación lógica para un dispositivo físico, y describe cómo deben los datos ser enviados y manejados por él.

Por ejemplo, una entrada de `printcap` definirá qué puerto vamos a usar, qué directorio tampón, qué proceso deben soportar los datos, qué clase de errores debemos notificar, qué volumen de datos se permiten enviar como máximo, o limitar el acceso de ciertos dispositivos.

Además, podemos definir distintos modos de procesar datos para una misma impresora. Por ejemplo, una misma impresora de HP puede manejar datos en formatos PostScript, HPGL y PCL, dependiendo de la secuencia de órdenes que le enviamos al comienzo de cada trabajo. Tendría sentido definir los tres modos de trabajo como sendas impresoras, cada una de las cuales procesará los datos dependiendo del modo de trabajo. Los programas que generan datos ps se enviarán a la impresora PS, los dibujos HPGL a la impresora HPGL, y así sucesivamente.

Llamaremos "filtros" a los programas que procesan los datos a imprimir. Un filtro puede incluso no enviar ningún dato al puerto.

4.6.1 Los campos de `/etc/printcap`.

Un ejemplo típico de entrada en `/etc/printcap` podría ser:

```
# Ejemplo de printcap con dos alias
impresora|HP850C:\
# lp es el dispositivo de impresion, en este caso, la primera impresora
:lp=/dev/lp1:\
# sd indica el directorio tampon
:sd=/var/spool/lpd/HP850C:
```

Como vemos, cada entrada de `/etc/printcap` se estructura en una serie de campos, encabezados por una etiqueta, y limitados por dos puntos, a excepción del primer campo, que describe la impresora. Los campos pueden tener tres tipos de valores - Texto, lógico y numérico, en los que nos extenderemos más adelante.

La primera línea de la entrada determina el nombre y alias de la impresora. La impresora por defecto debería llamarse "`lp`"; por ejemplo, si la impresora del ejemplo anterior es la única que tenemos, la primera línea sería:

```
# Ejemplo para la impresora por defecto
lp|HP850C:\
```

Podemos usar el nombre que nos apetezca como "*La Picadora de papel del despacho de Gertrudis*", aunque no parezca quizá muy práctico. Ojo: Sólo podemos tener una impresora llamada "`lp`".

Los siguientes campos son los más comunes, y los más importantes:

```
Campo   Tipo      Descripcion

lp      Texto     Dispositivo de impresion ( ej.: /dev/lp1 )
sd      Texto     Nombre del directorio tampon de esta impresora
lf      Texto     Fichero que almacena el registro de errores
if      Texto     Nombre del filtro de entrada
rm      Texto     Nombre del host de impresion remota
```

```
rp      Texto       Nombre de la impresora remota
sh      Logico      Suprime las paginas que separan los trabajos
sf      Logico      Suprime las paginas en blanco al final del trabajo
mx      numerico    Tamagnio maximo del trabajo de impresion (en bloques)
```

- **lp** apunta al puerto/dispositivo de impresión. Si especificamos `/dev/null` como dispositivo, el resto de los procesos se ejecutan normalmente, pero los datos de salida van a parar al inodoro. No se utiliza a excepción de las pruebas de configuración del dispositivo. Cuando configure una impresora remota con los campos `rm` y `rp` , debería poner ”`:lp=:`” en `/etc/printcap`, indicando que no está asignada a ningún dispositivo local.. No deje este campo en blanco a menos que use una impresora remota, o el demonio de impresión se quejará amargamente, si no especifica un dispositivo de impresión.

- **lf** guarda un registro de los errores de impresión. Cualquier fichero que especifique deberá existir antes de su uso, o no se registrarán las incidencias.

- **if** indica el nombre del filtro de impresión a ejecutar. Los filtros son programas que reciben datos por la entrada estándar, y los devuelven procesados por la salida estándar.

  Un empleo típico es detectar texto llano y convertirlo en PostScript para imprimirlo en ese tipo de trastos. Son muy útiles, por ejemplo, para eliminar el efecto escalera, o cambiar la página de códigos sin necesidad de cambiar la configuración de la impresora cada vez que la usemos entre UNIX y DOS.

  Cuando se especifica un filtro de entrada, el demonio de impresión no envía los datos pendientes al dispositivo especificado. En su lugar, arranca el filtro y asigna el fichero temporal a la entrada estándar, y el dispositivo de impresión como salida estándar.

- **rm y rp** controlan el destino de la impresión remota. Enviar el documento a imprimir a una impresora remota es tan fácil como indicar el anfitrión en `rm`, la impresora correspondiente en `rp`, y asegurarse que `lp` está vacío. Fíjese en que los datos se tamponan localmente antes de ser enviados, y que le ejecutará cualquier filtro que especifique.

  Una entrada típica de `/etc/printcap` en la máquina local (`pera.huerta.net`) para trabajar sobre la impresora picapapel, en la estación `rábano.huerta.net` (remota), sería:

  `picapapel:lp=:rm=rábano.huerta.net:rp=picapapel:sd=/var/spool/lpd/picapapel:`

  En la máquina remota necesitará que `/etc/hosts.equiv` o `/etc/hosts.lpd` contenga la línea `pera.huerta.net`; Tenga cuidado con los permisos de acceso. Vea el punto 4.6.4.

- **sh y sf**: Portadillas y separadores. Salvo que haya mucha gente distinta usando su impresora, probablemente no estará interesado en las páginas separadoras de trabajos. Las páginas de fin de trabajo son particularmente molestas cuando se trabaja con procesadores de texto, que componen páginas completas, por lo que si especificamos `sf`, tendremos al final de cada trabajo una página en blanco.

  sf es muy útil, sin embargo, si usamos la impresora para listar directorios, ficheros en crudo ..., asegurándonos que el trabajo sale completo de la impresión. Se puede presentar un problema si tenemos una impresora PostScript, al quedar residente el último tipo de letra utilizado. Con el campo `:tr:` lo evitaremos. Es preferible en estos casos dejar `:sh:`, y que sean los filtros quienes se encarguen de generar las portadillas.

- **mx** limita el tamaño del tampón de datos, señalando una cantidad en bloques de 1 K (1024 bits). Si `mx`=0, el límite viene dado por el espacio disponible en disco. Recuerde que lo que limitamos es el tamaño del tampón, no la cantidad de datos impresos.

  Si intentamos sobrepasar el límite, el fichero simplemente se trunca, y el usuario recibe el mensaje

  ```
  lpr: fichero: copy file is too large".
  ```

mx es útil si tiene programas que accidentalmente pueden generar un volumen desproporcionado de datos a imprimir (imágenes, por ejemplo); para impresoras ps, no suele tener mucho interés, pues un volumen pequeño de datos en formato ps pueden generar una notable cantidad de papel impreso.

Podemos sustituir el límite de `mx` escribiendo en cada directorio tampón un fichero llamado "`minfree`" que contiene el espacio mínimo disponible que permita aceptar trabajos, en forma de fichero texto con el número de bloques mínimos disponibles. Normalmente, suele ser un enlace con el original en `/var/spool/lpd`, ya que es inusual que cada impresora deba tener un mínimo diferente.

4.6.2 Probemos /etc/printcap con un ejemplo

El siguiente guión de shell es un filtro de entrada muy simple. Sólo encadena su entrada al final de un fichero en `/tmp`, tras una pancarta adecuada. Usaremos este filtro en el campo `if`, y enviaremos los datos a `/dev/null`, ahorrándonos las quejas del demonio especificando un dispositivo de impresión.

```
#!/bin/sh
# Este filtro debera colocarse en el directorio tampon, con el nombre
# filtro_entrada, propiedad de root, grupo lp y permisos -rwxr-xr-x
#
echo ------------------------------------------------>/tmp/testlp.out
date >> /tmp/testlp.out
echo ------------------------------------------------>>/tmp/testlp.out
cat
```

Y aquí tenemos nuestra flamante entrada en `/etc/printcap`. Vea que el formato es razonablemente inteligible, usando caracteres de continuación de línea "\" al final de cada una, excepto la última (de hecho, cada entrada en `/etc/printcap` es una sola línea):

```
impre|PLJ-H1998:\
:lp=/dev/null:\
:sd=/var/spool/lpd/PLJ-H1998:\
:lf=/var/spool/lpd/PLJ-H1998/errores:\
:if=/var/spool/lpd/PLJ-H1998/filtro_entrada:\
:mx#0:\
:sh:\
:sf:
```

Ojo: *NO DEJE ESPACIOS EN BLANCO*, o no funcionará (le aparecerán impresoras sin nombre, no logrará volcar los trabajos, y se acumularán en cola generosamente)

4.6.3 Valor y al toro: juntémoslo todo

- Primero: *DEBE SER EL SUPERUSUARIO* (`root`), tanto si le gusta como si no.
- Segundo: Compruebe los permisos y situaciones de `lpr`, `lprm`, `lpc`, `lpq` y `lpd`.
- Tercero: Cree el directorio tampón de su impresora: (recuerde que el propietario debe ser `root`, el grupo `lp`, y los permisos `-rwxrwxr-x` con las órdenes:

  ```
  mkdir /var/spool/lpd /var/spool/lpd/impre
  chown root.lp /var/spool/lpd /var/spool/lpd/impre
  chmod ug=rwx,o=rx /var/spool/lpd /var/spool/lpd/impre
  ```

- Cuarto: En el directorio tampón cree los ficheros necesarios, con los debidos permisos y propietarios:

```
cd /var/spool/lpd/impre
touch .seq errores status lock
chown root.lp .seq errs status lock
chmod ug=rw,o=r .seq errs status lock
```

- Quinto: Cree el script filtro de entrada en el directorio tampón. Use por ahora el filtro de prueba anterior. Asegúrese que el fichero pertenece a `root.lp`, y es ejecutable para todos.
- Sexto: Cree el fichero `/etc/printcap` tal y como lo hemos descrito en el punto anterior. Su propietario será `root` y sus permisos `-rw-r--r--`.
- Séptimo: Compruebe que `rc.local` (normalmente en `/etc/rc.d/`) contiene la línea `/sbin/lpd`, y añádala si no está. Esto hace que arranque el demonio de impresión al arrancar. De todos modos puede arrancarlo a mano con la orden `lpd`.
- Octavo: Haga una prueba: cruce los dedos y teclee

  ```
  ls -l | lpr -Pimpre
  ```

- Noveno: Busque en `/tmp` un fichero llamado `testlp.out`. Debería contener un listado del directorio, con un encabezado.
- Décimo: Si ha funcionado (y no dudamos que habrá sido así), edite `/etc/printcap`, y copie la entrada de prueba en el mismo fichero; En la primera entrada, cambie el nombre de la impresora a "`testlp`", o el que usted prefiera, pero que no use como nombre real de impresora. En la segunda entrada (que ahora será la de su impresora real), cambie el contenido de `lp=/dev/null` al del puerto de impresora (normalmente será `/dev/lp1`) de su ordenador, salvo que vaya a utilizar una impresora remota, en cuyo caso deberá definir `rm` y `rp`. Cambie el nombre del filtro `if` si tiene ya uno previsto, o suprímalo si no va a utilizar ninguno.
- Once: Reinicie el sistema, o mate el demonio de impresión y arránquelo, ya que `lpd` sólo lee `/etc/printcap` al comenzar su trabajo.
- Doce: Haga una prueba, y aprecie lo bien que funciona su impresora (Si tiene problemas con el efecto escalera, siga leyendo).

Para añadir nuevas impresoras, sólo tendrá que repetir la entrada de `printcap`, con las modificaciones pertinentes a cada dispositivo.

4.6.4 Y de la impresión remota, ¿qué?

Como primer paso, cualquier máquina que intente imprimir en su sistema, debe estar registrada en cualquiera de los ficheros `/etc/hosts.equiv` o `/etc/hosts.lpd`, que son simples ficheros de texto, con un nombre de maquina por línea. Es preferible el segundo, reservando el primero para proporcionar mayores permisos de acceso, lo que debería ser evitado en lo posible.

Puede restringirse el uso tanto por nombre como por grupos. Especifique los grupos autorizados con uno o varios campos "`:rg:`" en `/etc/printcap` - `:rg=admin:` sólo autorizará el acceso a la impresora a los usuarios asignados al grupo `admin`. Puede también limitar el acceso a aquellos usuarios con cuenta en su sistema utilizando el campo lógico `:rs:`.

4.6.5 Socorro: no funciona.

Algo falla; Lo primero que debemos hacer es revisar la existencia y los permisos y propiedad de los ficheros y directorios anteriormente descritos. Como comprobará los permisos de UNIX son algo juguetones hasta que se les controla. No se azore, y siga la siguiente rutina de control. Recuerde que cualquier modificación de /etc/printcap acarrea matar el demonio de impresión y arrancarlo de nuevo, o (último y brutal recurso) reiniciar el sistema.

1. ejecute "ls -l > /dev/lp?" (como root)
   - La impresora funciona: 2
   - La impresora no funciona: Compruebe el puerto de impresión, el cable, y que la impresora física este encendida y lista.
2. pruebe con lpr -P<impresora>
   - La impresora responde: 7
   - La impresora no responde: 3
3. Aparecen mensajes de error indicando que no se puede escribir .seq o que no se puede crear la cola: Compruebe que haya sitio en el disco, que el directorio tampón existe y que tiene los permisos necesarios.
   - no hay mensajes: 4
4. ejecute

   ```
   ps -ax | grep lpd
   ```

   y vea si aparece lpd en el listado, o hay un mensaje acerca del demonio de impresión:
   - lpd no está arrancado: Vea si rc.local tiene una línea /sbin/lpd; si no, añádala y reinicie el sistema. Puede arrancarlo a mano con /sbin/lpd como root.
   - lpd está funcionando: 5
5. Compruebe que el filtro de impresión especificado en if funciona. (filtro < datos >salida)
   - El filtro no funciona: corríjalo, y pruebe de nuevo
   - El filtro funciona: 6
6. Ejecute "lpc status".
   - Aparecen más impresoras que las descritas, algunas con el nombre en blanco: Hay espacios en blanco dentro de la entrada de la impresora en /etc/printcap. Revise la sintaxis de la entrada.
   - sólo aparece una impresora que no es lp: Si no ha especificado nombre de impresora, lpr usará lp por defecto. Pruebe

     ```
     lp -P<impresora>
     ```

   - La impresora está deseleccionada (down): ejecute

     ```
     lpc up <impresora>
     ```

     (debe ser superusuario).
7. La impresora funciona, pero imprime mal:
   - Se imprimen las cuatro o cinco primeras líneas, en escalera: Necesita un filtro de entrada. Vea 4.6.7
   - El fichero no se imprime completo: Elimine el límite en mx con :mx#0: en /etc/printcap.

Si tras este rastreo sus dificultades continúan, tenemos un buen problema. Una buena idea para aislar fallos es incluir el demonio "`syslogd`" en el arranque, de modo que obtengamos un registro de las incidencias de los procesos arrancados, que puedan ayudar al rastreo.

Un problema que ha ido apareciendo y desapareciendo de modo recurrente en distintas versiones de Linux está asociado a los mensajes "*job queued, but cannot start daemon*" o "*lpc: connect: no such file or directory*", y `lpd` sí está corriendo, parece manifestar un conflicto entre los módulos de red y el resto del sistema, y que se manifiesta si la red no está correctamente configurada. Si no está conectado a una red, suele corregirse añadiendo en la configuración de arranque:

```
ifconfig lo localhost
route add localhost
```

4.6.6 ¿Dónde puedo encontrar un `printcap` para la impresora xxxxx?

Si ha leído todo hasta ahora, comprobará que esta pregunta carece de sentido. No se le ocurra preguntar en usenet, si quiere que su dignidad se mantenga a salvo, y léase este documento completo.

4.6.7 El "efecto escalera", bajo control.

Mientras que en *DOS el indicador de nueva línea se compone de los caracteres CR y LF (retorno de carro y salto de línea), UNIX termina las líneas con LF. Esto hace que, si en una impresora configurada para *DOS imprimimos un documento UNIX, obtendremos una sucesión de líneas, que acabarán marchándose inmediatamente por el margen derecho del papel.

Tenemos dos vías para corregirlo: Reconfigurar la impresora para que interprete LF como $CR{+}LF$, algo molesto si alterna entre *DOS y Linux, o añadir un filtro de impresión, que lo haga sobre la marcha.

Ofrecemos dos: uno general, y el otro específico para impresoras HP.

```
#!/bin/sh
# filtro general para prevenir el efecto escalera
if [ "$1" = -c ] ; then
cat
else
sed -e s/$/^M/
fi
echo -ne \\f
```

```
#!/bin/sh
# filtro para corregir el efecto escalera en impresoras HP
echo -ne \\033\\&k2G
cat
echo -ne \\f
```

El carácter

```
^M
```

es el correspondiente a CR. Para obtenerlo con `vi` use *C-v C-m*, y con `emacs`, *C-q C-m* (C es la tecla *Control*).

El mejor sitio para guardar estos filtros es el propio directorio tampón de la impresora. Recuerde que deben especificarse en `/etc/printcap` en el campo `:if=<filtro>:`, con permisos `-rwxrwxr-x`, y propiedad `root.lp`.

4.6.8 Imprimir por el puerto serie es fácil.

Primero hemos de ver si `lpd` se queja enviando un error acerca de `ioctl(TIOEXCL)`. Si es así, deberíamos conseguir una versión de `lpd` que lo acepte (`lpd-590p2`, por ejemplo).

Necesitaremos configurar dos grupos de campos, además de la velocidad de transferencia `br` (Observación: Configurar el campo "`:fc:`" puede puentear el valor del campo "`:br#:`", así que asegúrese de configurar los dos correctamente).

Cada campo puede tener bits activados y desactivados. Se debe empezar desactivándolos con los campos "`:fc#*:`" y "`:xc#*:`", para activarlos posteriormente con "`:fs#*:`" y "`:xs#*:`".

Configurar `br#`; es obvio: "`:br#9600:`

Es muy fácil traducir la configuración de `stty` a los campos de `/etc/printcap`. Consulte la página de manual de stty para obtener más información.

Utilice `stty` para configurar el puerto de su impresora de manera que con "`cat`" pueda enviarle un fichero, e imprimirlo sin problemas. Aquí tiene un ejemplo de la configuración del puerto serie de la impresora:

```
# darkstar:/var/spool/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
```

Deberá estudiar el control de flujo de su impresora serie, y aplicar las correcciones necesarias a la configuración del puerto al que está conectado el equipo.

Una vez tengamos la configuración ajustada, de modo que con la orden

```
cat fichero > /dev/ttyS?
```

la impresora escriba correctamente, consultaremos el fichero `/usr/src/linux/include/linux/termios.h` (no será mala idea imprimirlo; así sabremos si funciona `:-)`.

En la sección que empieza con

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

está detallado el valor de los bits de "`fc#`" y "`fs#`". Apreciará que los nombres que aparecen aquí (tras las velocidades de transmisión) coinciden con una de las líneas del listado producido por `stty`. ¿Ve qué fácil?

Estos campos, en `stty`, van precedidos por un guión (`-`). Sume los valores (en octal) que aparecen en el listado. Este valor representa los bits a LIMPIAR; es el valor del campo `:fc#*:`. Note que, como deberá activar posteriormente los bit necesarios, podría usar directamente "`:fc#0177777:`".

Ahora repita el mismo proceso para las opciones sin guión en la salida de `stty`. En el ejemplo, los bits importantes son `CS8 (0000060)`, `HUPCL (0002000)` y `CREAD (0000200)`. Considere también los indicadores para la velocidad de transmisión (en mi caso `0000015`). Súmelos; en este caso, tendremos `0002275`: el valor del campo `:fs#*:` (En mi caso, "`:fs#02275:`" funciona).

Haremos lo mismo con la siguiente sección de `termios.h`, que comienza con "`c_lflag bits`". En mi caso, no he necesitado ningún ajuste, y utilizo "`:xc#0157777:` y "`:xs#0:`".

Una vez modificado `printcap`, intente imprimir con `lpr`. Si no funciona, siga leyendo.

4.6.9 `cat` funciona, pero `lpd` no.

Mantener "`lpd`" en forma se explica en la página adecuada del manual, pero si le persiguen los problemas de configuración, puede evitar que `lpd` establezca la configuración del puerto, haciendo que su impresora no presente un dispositivo normal (vea la sección 4.6.10).

El proceso a seguir es el siguiente:

1. el campo `:lp:` de `/etc/printcap` deberá apuntar a `/dev/null1` (créelo con `mknod /dev/null1 c 1 3`), porque no queremos a `/dev/null` sólo para nosotros. Elimine los campos `:br:` y `:fs:`, `:fc:`, `:xs:` y `:xc:`.
2. Cree un script, por ejemplo, `testserie`, tal que este:

   ```
   #!/bin/sh
   echo if: $* >> /var/spool/lpd/resultados
   # /dev/lp debe estar enlazado (con ln -s) a /dev/ttyS?, donde esta la
   # impresora
   exec su-filtro-de-entrada-viejo $* > /dev/lp
   ```

   o si no tiene un `:if:` instalado

   ```
   #!/bin/sh
   echo if: $* >> /var/spool/lpd/resultados
   # asumimos que /bin/sh es bash al usar "echo -ne"
   echo -ne \\f > /dev/lp
   ```

3. Asegúrese que es ejecutable y legible para todos (`rwxrwxr-x`). Pruebe el filtro con

   ```
   # testserie < algo-que-imprimir
   ```

   y vea si sale algo por la impresora. Modifique `/etc/printcap`, para que `if` sea "`testserie`" (o el nombre que le haya dado).
4. Con `stty`, configure el puerto de la impresora. Intente imprimir. Ahora podrá saber si hay datos en el tampón, y algo *debería* imprimirse si el paso 3 funcionó.

Ahora, si el proceso con `if` funciona, y cree que la configuración del `printcap` original es la correcta, vuelva a revisar la configuración del puerto, las etiquetas de `termios.h` ... Si aún no funciona, probablemente `lpd` tiene un defecto, y deba obtener una versión nueva.

4.6.10 Impresoras que no son dispositivos simples.

Piense en una impresora conectada de algún modo tortuoso a nuestra red ... —agárrese: está enchufada a un ordenador con el que sólo podemos comunicarnos mediante correo electrónico—

Para imprimir a través de `lpr`, el campo `:lp:` debería ser redirigido a un dispositivo nulo,

```
mknod /dev/null1 c 1 3
```

(no a `/dev/null`, pues `lpd` lo usa en exclusiva). El filtro de entrada (`:if:`, recuerde) debe explícitamente codificar y enviar como correo al ordenador remoto su salida.

Se rumorea que alguien consiguió con algo similar con Novell NetWare y el agente de correo Charon; también hay quien cree en los Reyes Magos.

4.7 Trucos y problemas

4.7.1 `lpr -i` o como modificar la impresión a nuestro antojo.

La opción `-i` envía argumentos al filtro de impresión. Ante todo, necesitamos un filtro instalado en `printcap` para que la opción '`-i`' funcione. Ésta se limita a pasar a través de `lpd` hacia el filtro. Si quisiera un filtro que permitiera configurar su impresora sobre la marcha, podríamos utilizar un filtro como este:

```
#!/bin/sh
# Mi programa de configuracion se llama confimpre.
exec /usr/lib/confimpre $*
```

que desplace el margen izquierdo, y pasar como argumento el margen deseado.

```
#!/usr/bin/perl
# confimpre: Desplaza el margen izquierdo en un texto ascii
# Usamos perl porque convertir numeros a caracteres
# es un campo de minas en programacion shell
for  ( $i=0 ; !($_ = $ARGV[$i]) || !/^-i([0-9])+/;
$i++) {}
print pack( "cAc" , 27 , "l" , $1) ;
while (<STDIN>) { print; }
```

4.7.2 '`lpc`' y '`lpq`' advierten : "`missing daemons`"

Un proceso `lpd` corre continuamente, y se desdobla en hijos para manipular cada impresora en tanto se necesita. La salud del proceso principal no se manifiesta explícitamente con '`lpc`'; basta con la ausencia de errores. El comando '`lpc stat`' mostrará de manera normal el mensaje "`no daemon present`" para cada cola actualmente inactiva. Si la impresión se desactiva, o la cola se vacía, '`lpq`' es bastante más escandaloso, gritando "`Warning:  no daemon present`", aunque, de hecho, no hay condición de error. Si el demonio está ausente cuando hay entradas en la cola, y no ha sido explícitamente detenido, probablemente se debe a un fallo en algún filtro. Fíjelo y reinicie con "`lpc up` <`cola`>.

De vez en cuando, al desactivar una impresora, '`lpc`' alucina, e intenta matar demonios inexistentes, emitiendo un surtido de irritantes e inofensivos mensajes de error. Esta "característica" es muy rara en versiones posteriores a `lpd-590p2`.

4.7.3 Impresión sobre red (notas adicionales)

Dado que la máquina que alberga la impresora física debe saber quien se conecta, deberemos dar de alta al resto de los ordenadores en los ficheros `/etc/hosts.equiv` y `/etc/hosts.lpd`, con sus nombres canónicos, o direcciones IP. Si no está seguro del nombre completo de una máquina, puede incluir todos sus alias conocidos.

Por motivos de seguridad, es preferible que las máquinas que sólo deban tener acceso a la impresora queden restringidas a `/etc/hosts.lpd`.

Si el servidor de impresión no ejecuta un sistema de impresión tipo BSD (*UnixWare*, *HP-UX* versión 10 ,...), debería ser posible trabajar con él, teniendo en cuenta que los ficheros de autorización pueden variar notablemente, cambiando en cada sistema.

Si aún no puede imprimir sobre un sistema distinto, queda un as en la manga: simplemente use una orden remota, tal que `rsh`, `rexec` ...

```
# rsh rabano.huerta.net "lp -dlp" < fichero
```

permitirá imprimir sobre un SYSV, desde nuestra `pera.huerta.net`.

4.8 Los filtros de impresión

Los filtros en UNIX, son meramente programas (y como tales deben tener permisos de ejecución) que leen la entrada estándar y escriben sobre la salida estándar, independientemente de su complejidad.

Los filtros de `lpd` lo son en el sentido que leen STDIN y escriben en STDOUT; su peculiaridad estriba en que deben asumir que la entrada estándar es un fichero de acceso aleatorio, sobre el que se pueden realizar operaciones con `lseek()`.

Un filtro que interpreta PostScript nos servirá de ejemplo:

```
#!/bin/sh
/usr/bin/gs -q -dSAFER -dNOPAUSE -r?? -sDevice=? -sOutputFile=- - < /dev/null
```

La primera línea invoca al intérprete de comandos que lo interpretará. Podemos usar `csh`, `bash`, `perl`, `tcl`, etc...; va por gustos.

En la segunda línea usamos `gs` como filtro, indicando que la entrada es STDIN (el guión del final), y la salida, STDOUT (`-sOutputFile=-`). Lea la página de manual de '`gs`' para más detalles.

Es buena práctica guardar los filtros de entrada en el directorio tampón de la impresora en la que se usan (o en `/var/spool/lpd/`), aunque las buenas maneras recomienden camas separadas para datos y programas.

Más difícil aún:

```
#!/bin/sh
/usr/tex/bin/dvips -f | /usr/bin/gs [...] -sOutputFile=- -
```

Como se puede ver, este filtro procesará un fichero texto de formato TEX o LATEX, obteniendo un fichero `.dvi`, que es transformado en PostScript, y reprocesado con ghostscript para imprimirlo en una impresora normal.

Ojo: Si usa impresión remota con `lpr`, el filtro deberá encontrarse en la máquina remota; `LPRng` obvia este inconveniente.

4.8.1 Los filtros de salida

Son como el `goto` en 'C', se pueden usar, pero realmente, es difícil encontrar aplicaciones para ellos. Dado que cualquier comando de iniciación de la impresora es ignorado, si tenemos un trabajo rodando que cambia algo (el tipo de letra, por ejemplo), el nuevo trabajo probablemente acabará mal impreso. No los use.

4.8.2 Depuración de filtros

Es absolutamente trivial: `mifiltro <fichero >/dev/lp1` (como `root`). Comprobaremos de inmediato si de verdad funciona.

Un truco muy útil para los filtros con argumentos en línea de comandos consiste en incluir la línea

```
echo $* >> /tmp/filter-log
```

cerca del principio del guión. Recuerde incluir en la primera línea '`#!/bin/sh`', asegúrese de que el propietario es `root`, y que son ejecutables por todo el mundo.

4.8.3 Filtros Mágicos

Tienen de especial estos programas que son capaces de identificar la familia de los datos a imprimir a partir de patrones característicos de datos en situaciones determinadas. Son generalmente guiones Shell, Perl, programas en C ...; el programa identifica la fuente de datos, y llama a un filtro ordinario que lo procesa de modo adecuado. Con un filtro mágico adecuado y tres filtros no mágicos podemos imprimir cosas como esta:

```
# lpr -d fich1.dvi fich2.dvi.Z fich3.ps fich4.tex.gz
```

Un buen sitio para encontrar filtros mágicos como el del ejemplo anterior es:

`ftp://tsx-11.mit.edu/pub/linux/sources/usr.bin/magic-filter-??.tar.gz`.

Como podrá suponer, nunca se deberían usar como filtros de salida.

Un ejemplo: el siguiente filtro mágico identifica un fichero fuente ps o texto, y lo procesa de acuerdo a su clase:

```
#!/bin/sh
printf "<Filtro magico para imprimir texto y ps.>"
read first_line
first_two_chars=`expr $first_line : '\(..\)'`
if [ "$first_two_chars" = "%!" ] ; then # Documento PostScript.
/usr/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=??? -sOutputFile=- - < /dev/null
else
# Texto llano - Si hace falta, podemos incluir aqui la
# correccion del efecto escalera
echo -n $first_line
cat
printf "\014"
fi
```

Para desazón de muchos, habrá ya notado que un script ejecutado por alguien distinto al propietario puede ser una brecha de seguridad. Este no es el caso de los filtros de `lpd`, ya que el entorno de los guiones no está al alcance de potenciales piratas.

4.8.4 Los usuarios de HP estamos de suerte

Hasta donde hemos podido averiguar, HP no tiene ningún programa que permita configurar sobre la marcha sus impresoras (¡¡ ni siquiera para sus propios sistemas UNIX!!), salvo que use MS-Windows, o *DOS; por lo que sólo nos queda, bien usar DOSEMU, o salir a DOS, configurar la impresora, y volver a Linux.

Existe un hermoso programa llamado `hpmodeset`[4], escrito en Eiffel, que hace el trabajo por nosotros, muy fácil de usar. Su autor es Glenn Maughan (`glennm@insect.sd.monash.edu.au`); se puede conseguir en `ftp://hornet.sd.monash.edu.au/pub/hpmodeset/v2.0`.

Permite fijar economode, definición, tipo de papel, etc... en ficheros PCL (el lenguaje de las impresoras HP) y ps. No es tan espectacular como el programa de windows, pero funciona muy bien, y es susceptible de ser utilizado como filtro de impresión.

[4]Nota del Traductor:
Hay también otro programa más moderno y práctico, llamado `hpset`, localizable en cualquier *mirror* de Sunsite, en `../system/printing/`

5 ¿CÓMO SE IMPRIME ... ?

Ya sabemos con qué imprimimos bajo Linux. Ahora haremos una visita a los programas más normales para generar el texto a imprimir usualmente disponibles. Advierta que la mayor parte de los programas de impresión bajo otros UNIX pueden ser muy fácilmente adaptados a Linux.

5.1 Texto Llano

Bajo Linux, (como cualquier otro U*X), lo más fácil es imprimir texto llano. Cualquier editor estándar de UNIX (`vi`, `emacs`), o alguno más sofisticado (`nedit`, `jed`, ..) le producirán tantas arrobas de letras como desee.

Además, la mayor parte de los programas que a continuación se describen son capaces de generar documentos en texto llano, con las órdenes adecuadas.

5.1.1 pr

La mayoría de los ficheros de texto llano en U*X tienen tendencia a ser un mar de caracteres, sin saltos de página ni nada que permita un documento mínimamente presentable.

`pr` surgió para obviar este inconveniente. Procesando el documento a través de `pr`, se le añaden cabeceras y/o pies, números de página, márgenes superior e inferior... Como cualquier otra utilidad UNIX, `pr` tiene dos o tres millones de opciones, debidamente reseñadas en la página del manual.

5.2 PostScript

Si no tiene que imprimir texto llano, es muy probable que tenga que imprimir PostScript. Si su impresora lo interpreta, sus problemas terminan aquí; si está en el caso contrario, situación normal para casi todos, la cosa se complica un poco, pues necesitará un programa que lea PS y lo traduzca al lenguaje de su impresora.

5.2.1 Ghostscript

El modo normal de interpretar ps bajo Linux es usar ghostscript, que , como casi todas la utilidades, procede del proyecto GNU. Herramienta versátil donde las haya, `gs` traduce PostScript a formato X Window, impresoras IBM, Epson, HP, Canon ..., fax, y casi para cualquier dispositivo que se le ocurra. Los dispositivos soportados en cada versión de ghostscript están debidamente relacionados en la documentación del programa; puede obtenerse una relación con la orden '`gs -help`'.

`gs` se ha convertido en utilidad estándar en la mayoría de las distribuciones de Linux, que la incluyen completa. Advierta que los ejecutables binarios pueden necesitar '`libX.so.???`' tal como vienen en el controlador de pantalla X Window.

La distribución ºficial" en forma de ficheros fuente, que deben ser compilados en su sistema puede encontrarse en:

`ftp://prep.ai.mit.edu/pub/gnu/ghostscript-*.tar.gz`

`ftp://prep.ai.mit.edu/pub/gnu/ghostscript-fonts-*.tar.gz`

La instalación mínima de los binarios de `gs` y algunos otros programas necesarios para imprimir la documentación de Linux está disponible en

`ftp://sunsite.unc.edu/pub/Linux/apps/tex/texmin/texmin-*.tar.z`

Esta distribución no contiene ningún tipo de letra PostScript (ni las necesitará para imprimir ficheros `.dvi`).

La documentación principal de ghostscript se encuentra en el fichero `use.doc`, en el directorio de fuentes, o en `/usr/lib/ghostscript/doc/` si no tiene los fuentes.

Para imprimir ps determine primero el nombre de su dispositivo con la orden '`gs -help`', que lista los dispositivos instalados. Si el suyo no está en la lista, deberá recompilar `gs` (No se asuste, siga las instrucciones que figuran en `make.doc`. Necesitará libres 5 o 6 Mb de espacio en disco). Con la orden:

```
# gs -dNOPAUSE - sDEVICE=???? -sOutputFile=/dev/???? fichero.ps
```

el documento aparecerá (eso esperamos) en su impresora sin problemas. Como podrá imaginar, a mayor versión, más dispositivos soportados. En la documentación incorporada se describe el procedimiento para desarrollar nuevos dispositivos si fuera necesario.

`gs` puede imprimir en la mayoría de las resoluciones soportadas por su impresora; '`-r300`' '`-r150`', '`-r360x180`' son ejemplos de las opciones para su control. Las impresoras matriciales necesitan que se fije esta opción, al no soportar normalmente la resolución por defecto de 300x300.

5.3 Mi impresora sólo trabaja con PostScript

Si su impresora sólo trabaja con PostScript, o quiere ahorrarse andar configurando el dispositivo cada vez que tiene que cambiar el tipo de documento a imprimir, hay toda una familia de programas que transforman el texto llano a PostScript.

5.3.1 mpage

Transforma texto llano a PostScript y/o imprime más de una página en cada hoja de papel a partir de PostScript o texto. Puede encontrarlo en

`ftp://wuarchive.wustl.edu/pub/mirrors/unix-c/PostScript/mpage-tar-z`

el sufijo `-z` equivale a `.Z` (`compress`). La orden

```
man -t <orden> | mpage
```

enviará una copia de la página en formato PostScript a `lpr`.

5.3.2 a2ps, enscript, nenscript

A partir de un texto ascii llano, `a2ps` produce una página debidamente formateada, con cabeceras, pies, números de página ... Se encuentra en el mismo lugar que `mpage`.

`enscript` y `nenscript` hacen lo mismo que `a2ps`. Puede encontrar `nenscript` en:

`ftp://sunsite.unc.edu/pub/Linux/system/Printing/nenscript*`[5]

[5]Nota del Traductor:
Ojo; Aquí podemos tener problemas con los caracteres extendidos; por ejemplo: `a2ps` (la versión que tengo, al menos) genera un fichero `ps` que no tiene acentos, eñes .., al no definir una tabla tipográfica para ellos. En cambio, `nenscript` si lo hace.

5.3.3 gslp

Incluida en la distribución de ghostscript, `gslp` transforma un fichero texto a PostScript. Úselo con la orden:

```
# gs -q -sDEVICE=???? -dNOPAUSE -- gslp.ps fich.text
```

(el programa es `gslp.ps`, escrito en PostScript, que, en muchos aspectos, es un lenguaje de programación, más que un conjunto de órdenes de impresión). Hay un ejecutable en la distribución que evita dar estos rodeos.

5.3.4 Utilidades para PostScript

Existen cientos de pequeños programas que permiten modificar ficheros ps. Puede encontrarlos en

- `ftp://achilles.doc.ic.ac.uk/tex/inter/psutils/`
- `ftp://ftp.cs.psu.edu/pub/src/psutil.tar.gz`
- `ftp://ftp.uu.uunet`
- `ftp://wuarchive.wustl.edu`

5.4 TeX, LaTeX

Si hay algo cierto en el mundo de UNIX, especialmente a nivel académico, es que antes o después uno se acaba topando con Él. `TeX` es un sistema de formateado de textos[6]. Funciona de modo análogo a un compilador: El código fuente corre a través del programa tex, y obtenemos un fichero `.dvi`, independiente del dispositivo, que normalmente debe ser procesado de nuevo para obtener un documento impreso. La altísima calidad del producto obtenido justifica la complejidad del proceso.

El trabajo con `TeX`, no obstante, se ve gratamente suavizado mediante el uso de procesadores de texto, como `LyX`, con el que ha sido reelaborado este documento.

Remitimos al lector al libro "*LaTeX, a document preparation system*" de Leslie Lamport (ed. *Addison Wesley*) 2? edición, que analiza en detalle la versión más popular de este extraordinario sistema de tratamiento de textos.

5.4.1 dvips

Convierte el fichero dvi generado a partir de un documento `TeX` en un fichero PostScript, que puede imprimir a través de `gs`, o directamente en la máquina adecuada. La mayoría de las distribuciones de Linux incluyen este paquete.

La orden

```
dvips -f1 fich.dvi | lpr
```

es típica. `dvips` responde tanto a los argumentos de la línea de comandos como al fichero de configuración "`config.ps`", donde puede organizar las cosas para que la salida vaya directamente a la impresora, por ejemplo. Así, bastará escribir `dvips fichero.dvi` para obtener el documento final.

`dvips` tiene unas cuantas opciones interesantes; por ejemplo `dvips -r1 fich.dvi` imprimirá el documento en orden inverso. Los propietarios de impresoras HP apreciarán esta característica.

[6]Nota del revisor:
Más bien un *Procesador de Documentos*

5.4.2 `eps`, `dvilj`, `dvi500`

Transforman los ficheros dvi en los lenguajes de impresoras Epson y HP LaserJet y DeskJet respectivamente. Suelen estar incluidos en la mayor parte de las distribuciones, y, por supuesto, en `ftp://sunsite.unc.edu`

5.5 `texinfo`

Es el formato nativo del proyecto GNU. `emacs` puede ser forzado a producir un fichero de información desde TeXinfo, y TeX producirá un excelente resultado a partir de este mismo fichero. Realmente, el documento está en formato TeX, y necesita que el fichero de macros `texinfo.tex` esté instalado en su sistema. Simplemente ejecute '`tex fichero`' dos veces (con el fin de generar índices), y obtendrá un fichero dvi que podrá imprimir o visualizar a su placer.

Con `emacs` puede teclear además '`M-x texinfo-format-buffer`' para convertir el fichero texinfo en un fichero info legible con emacs '`M-x info`', u otro visor de su elección.

Hay otros programas que leen y formatean info desde un fichero texinfo. Están disponibles en `ftp://prep.ai.mit.edu/pub/gnu/`

5.6 `*roff`; páginas man

Si hay un generador de documentos virtualmente universal es `troff`, y sus primos cercanos, debido a que las páginas de manual se escriben para ellos. Las órdenes de formato de documentos serán ¿sorprendentemente? familiares para los (no se si aún queda alguno vivo) forofos de las versiones más populares de *Wordstar*.

La orden para imprimir las páginas de manual es

```
man -t <comando> | lpr
```

`groff` traduce la página a PostScript, que se procesa sin mayores problemas, como ya sabemos. Obtendremos así una página impresa muy aparente. Esto, sin embargo, depende mucho del programa '`man`' que viene con su sistema. Si el suyo no lo permite, consiga la versión escrita en `perl`, en: `ftp://sunsite.unc.edu/pub/Linux/system/Manual-pagers/`.

Está íntegramente escrito en `perl`, y es por tanto muy personalizable.

Otro camino consiste en encontrar el fichero fuente de `nroff` de la página en los directorios de `man`, e imprimirlo con la orden:

```
groff -fichman -Ttipo orden.1 | lpr
```

donde tipo puede ser '`ascii`', '`dvi`' '`ps`' '`X100`', '`X75`', '`latin8`'

5.7 Adobe Acrobat. Ficheros PDF

Estándar emergente donde los haya, los papás de PostScript han desarrollado un formato de documentación que permite (al menos eso venden) absoluta independencia del sistema operativo. Un fichero PDF se imprimirá igual de bien en un ordenador con MS-Windows, un Mac*, un CojoSuperOrdenador Cray, o un PC con Linux.

Los documentos PDF se traducen justo a la hora de imprimirlos al formato que cada sistema operativo requiera; en UNIX, a PostScript.

Puede obtenerse una copia del lector de Adobe en `http://www.adobe.com`.

Si prefiere algo menos "comercial", `Xpdf` será de su agrado. La última versión está en

`http://www.contrib.andrew.cmu.edu/usr/dn0o/xpdf/xpdf.html`

5.8 Tipos

(Esta sección contiene información no específica a los dispositivos). La información pertinente al dispositivo X11 de `gs` (y ghostview) se incluye en el apartado de visualización.

Todas las versiones de `gs` vienen provistas de un surtido de tipos de dominio público, muchas de ellas generadas a partir de mapas de bit, con baja calidad. Sin embargo, `gs` puede usar cualquier tipo PostScript clase 1 o 3 que tenga a mano. Por ejemplo, *Adobe Type Manager* (que no sea de Mac) incluye tipos utilizables. Sitúelos (normalmente `*.pc?`) en `lib/ghostscript/fonts/`, y añada al fichero `[...]/ghostscript/Fontmap`' líneas del estilo

```
/Courier            (com_______.pfb) ;
```

En el servidor de correo de impresión puede encontrar el fichero `fontmap.atm`, que contiene un *fontmap* de los tipos normalmente incluidos en *Adobe Type Manager*.

Puede encontrar tipos Adobe Type 1 en:

`ftp://ftp.cica.indiana.edu/pub/pc/win3/fonts`

`ftp://archive.umich.edu/msdos/mswindows/fonts`

En el fichero `comp.fonts FAQ`, en `ftp://rftm.mit.edu`, puede encontrar información útil acerca de tipos de impresión.

La conversión entre familias de tipos es delicada. `gs` incluye herramientas para generar tipos escalables a partir de mapas de bits (cuanto más grandes, mejor).

`groff` tiene también herramientas para usar tipos Tfm/mf (TeX) y pfb (type 1) en documentos *roff. X11R5 incluye herramientas procedentes de IBM para presentar tipos type 1, con sus correspondientes páginas de manual. Eche un ojo también al *logical fontutils* en `ftp://prep.ai.mit.edu/pub/gnu/`.

5.9 Documentos para fax.

Aunque no parezca el lugar más adecuado para su reseña, ¿qué hacemos al enviar un fax más que imprimir un fichero en un terminal remoto?. Mediante fax, puede transmitir documentos originalmente en PostScript, dvi, texto llano....

Para enviar un fax, el documento original debe ser transformado en una imagen TIFF del grupo III que será transmitida por el fax-módem. Todos los programas mencionados contienen utilidades que permiten transformar las distintas clases de documentos al formato necesario. Además, `gs` (también aquí) tiene dispositivos adecuados para ello (`-sDEVICE=tiffg3`).

Tenga en cuenta que hay dos juegos de órdenes que un fax puede soportar: Clase 1 y Clase 2. En el primer caso, hoy ya casi extintos, gran parte de la gestión de transmisión de datos debe hacerla el ordenador anfitrión vía el programa de fax. Esto hace que en condiciones de sincronización crítica, el control se vuelve casi imposible. Los fax de clase 2 son más caros, pero mucho menos propensos a dar guerra. No confunda la 'clase' y el 'grupo'. Deberá tener, por supuesto, un fax-módem de grupo III.

5.9.1 FLEXFAX

FlexFax está escrito en C++ (necesita instalar g++ para compilarlo), y soporta fax de clase 1 y 2. Utiliza `ghostview` (o cualquier otro visor de ps) . Puede ejecutarse en lugar de `getty`, y decide, según el tipo de llamada, si activar el fax, o presentar un login. Permite la emisión de faxes a través de una red, y tiene un zurillón de utilidades. Se configura vía '`configure`', y para un ordenador aislado es un tanto mastodóntico. La versión actual (4.0 beta 018, según escribo) puede encontrarse en: `http://www.vix.com/hylafax/`

5.9.2 `mgetty+sendfax`

`mgetty+sendfax` controla sólo fax de clase 2, y puede también actuar como contestador automático, si el módem lo soporta. Al igual que FlexFax, se instala en lugar de `getty`, controlando qué entra y sale. Muy adecuado si no necesitas trabajar sobre red. Un excelente conjunto de programas disponible en

`ftp://sunsite.unc.edu/pub/Linux/system/Serial/mgetty+sendfax*.tar.gz`

`ftp://tsx-11.mit.edu/pub/Linux/sources/sbin/mgetty+sendfax*`

`ftp://ftp.leo.org/pub/comp/networking/communication/modem/mgetty/`

5.9.3 EFAX

Controla fax de clase 1 y 2; no discrimina entre una conexión de datos y un fax, tiene su fuerte en su facilidad de uso. Necesita el paquete NetPBM (ftp.x.org). Utiliza `xv` para ver los fax recibidos. Lo encontrará en:

`ftp://sunsite.unc.edu/pub/Linux/apps/comm/efax*`

6 Visualización

Hubo un tiempo en que a alguien se le ocurrió el concepto de la ºficina sin papeles"; no haría falta manejar papel impreso durante el tránsito de documentos. Qué falta de vista. ¿Cómo podríamos si no fardar de impresora ?. Nos ha quedado de todo aquello una serie de programas que nos facilitan ver el resultado final de un trabajo sin su materialización.

6.1 Con X-Window instalado

6.1.1 `ghostview`

El complemento ideal de '`gs`', visualiza documentos PostScript en un sistema X Window. Permite elegir páginas individuales, o grupos de ellas para su posterior impresión. La distribución oficial está en

`ftp://prep.ai.mit.edu/pub/gnu/ghostview-*.tar.gz`

Fácil de construir, `ghostview` necesita `gs` para ser ejecutado. Las últimas versiones utilizan las fuentes estándar de X Window, facilitando la legibilidad en las condiciones de baja resolución de los monitores. La verdad es que funciona muy bien; de hecho, muchos piensan que mejor que el paquete `pageview` de Sun. Si lo desea, puede utilizar fuentes distintas a las del sistema (por ejemplo, Type 1), simplemente situando la orden

```
ghostscript.useExternalFonts:false
```

en su fichero `.Xdefaults`.

Puede obtenerse más información de la documentación que acompaña al programa, y en la correspondiente página de manual.

6.1.2 gspreview

Tenemos aquí otro interfaz para ghostscript. Más sencillo que `ghostview`, carece del millón de opciones de éste, pero trabaja razonablemente bien. Puede encontrarlo en

`ftp.x.org/contrib/gspreview*`

6.1.3 xdvi y xtex

El equivalente para documentos `.dvi` de `ghostview`, tiene una curiosa y útil "lupa" incorporada. Extremadamente legible, viene incluido con la mayoría de las distribuciones de TeX. Para ver un fichero basta con '`xdvi fich.dvi`'. Su situación usual es

`ftp://ftp.x.org/contrib`.

`xtex` tiene La misma utilidad que `xdvi`. Disponible en `ftp://ftp.x.org/contrib/xtex-*.tar.z`

6.2 ¿Y si no tengo instalado X Window ?

Instalar X puede ser una mala opción si su equipo no dispone de memoria suficiente (8 Mb recomendados para ello). No se preocupe, también hay solución.

6.2.1 Ghostscript, de nuevo

Ghostscript incluye en su distribución dispositivos para control del soporte físico de los PC; no es una buena solución para UN*X; Las últimas distribuciones, sin embargo, incluyen soporte para librerías VGA de Linux (`svgalib`). El dispositivo pertinente se llama 'Linux'; la orden

```
gs -sDEVICE=linux fichero.ps
```

mostrará el documento en pantalla. La variable de entorno `GSVGAMODE`, muy importante, fija el modo de vídeo deseado, según los valores que aparecen en el fichero de vgalib `vga.h`.

Si no lo tiene en su distribución, puede obtener el parche en el servidor de correo de impresión, o en

`ftp://ws105.zfn.uni-bremen.de/pub/gs261-linuxdriver.sh`

`ftp://ws105.zfn.uni-bremen.de/pub/gs261-svgalib.tar.gz`

`ftp://ftp.cdrom.com/pub/linux/misc`

y por supuesto, en `ftp://sunsite.unc.edu`.

(Pruebe mejor en los diversos duplicados de este lugar. Sunsite está normalmente saturado[7]).

6.2.2 dvgt

Permite la visualización de ficheros .dvi bajo la librería `svgalib`, o en diferentes clases de terminales gráficas, incluyendo vt, tek, o un PC corriendo MS-Kermit. Disponible en sunsite.

7 Copyright

Salvo otra disposición, los documentos COMO de Linux están registrados por sus autores respectivos. Los documentos COMO de Linux pueden ser reproducidos y distribuidos en todo o en parte, en cualquier medio físico o electrónico. Las traducciones y similares están permitidos sin necesidad de permiso expreso.

[7]Para los residentes en España, `ftp://sunsite.rediris.es` es un *mirror* inmejorable.

La redistribución comercial está permitida y recomendada; sin embargo a los autores les gustaría tener conocimiento de tales distribuciones.

Resumiendo: Nos gustaría promover la distribución de esta información a través del mayor número de canales posible. Sin embargo, nos gustaría retener los derechos de autor de los documentos COMO, y ser informados de cualquier plan para su redistribución.

Si tiene dudas, contacte con Greg Hankins[8] el coordinador de los COMO de Linux, en `greg.hankins@cc.gatech.edu` .

8 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el **INSFLUG** se orienta preferentemente a la traducción de documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. `:)`), etc.

Diríjase a la sede del INSFLUG para más información al respecto.

En la sede del INSFLUG encontrará siempre las **últimas** versiones de las traducciones: `www.insflug.org`. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

Francisco José Montilla, `pacopepe@insflug.org`.

[8] En cuanto a coordinación de traducciones al castellano, ver sección 8