

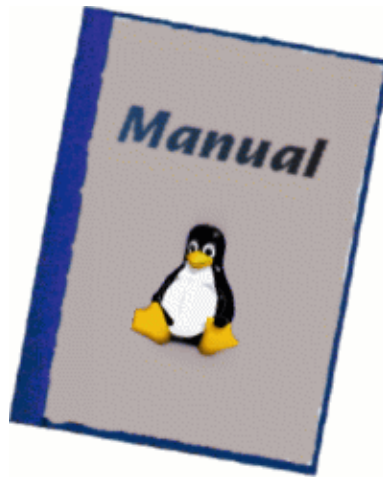
Escribiendo páginas de manual



by Guido Socher ([homepage](#))

About the author:

A Guido le gusta Linux porque es muy flexible y ofrece muchas más posibilidades que cualquier otro sistema operativo.



Abstract:

Todo buen programa que se pueda usar desde la shell de UNIX debería estar documentado en su propia página de manual. Este tutorial es una breve introducción a la creación de páginas de manual.

Introducción

A menudo la documentación es más importante que el propio software, especialmente cuando el autor no sea el único que va a utilizar dicho software. Incluso cuando escribo un programa que no pienso publicar también escribo su documentación, ya que varios meses después podría olvidar cómo usar el programa y con una buena documentación sabré en pocos segundos cómo hacerlo.

Las utilidades tradicionales de Linux en línea de comandos siempre se han documentado en páginas de manual. Un simple `man comando` nos dirá cómo utilizar el comando.

Las ventajas de las páginas de manual sobre otras formas de documentación son

1. Se pueden ver en pocos segundos en un terminal Linux
2. Se pueden convertir fácilmente a otros formatos: HTML, PDF, Postscript, Texto,...
3. Las páginas de manual no sólo se pueden ver en ventanas de terminal, sino también desde otros programas como konqueror (basta con teclear: `man:comando`)

Las secciones

Las páginas de manual están estructuradas en secciones, de la misma forma que un libro está estructurado en capítulos. Por ejemplo, existen dos páginas de manual de printf. Una para la función de biblioteca C (sección 3) y la otra para el comando de shell (sección 1):

```
> whichman -0 printf
/usr/share/man/man1/printf.1.bz2
/usr/share/man/man3/printf.3.bz2
```

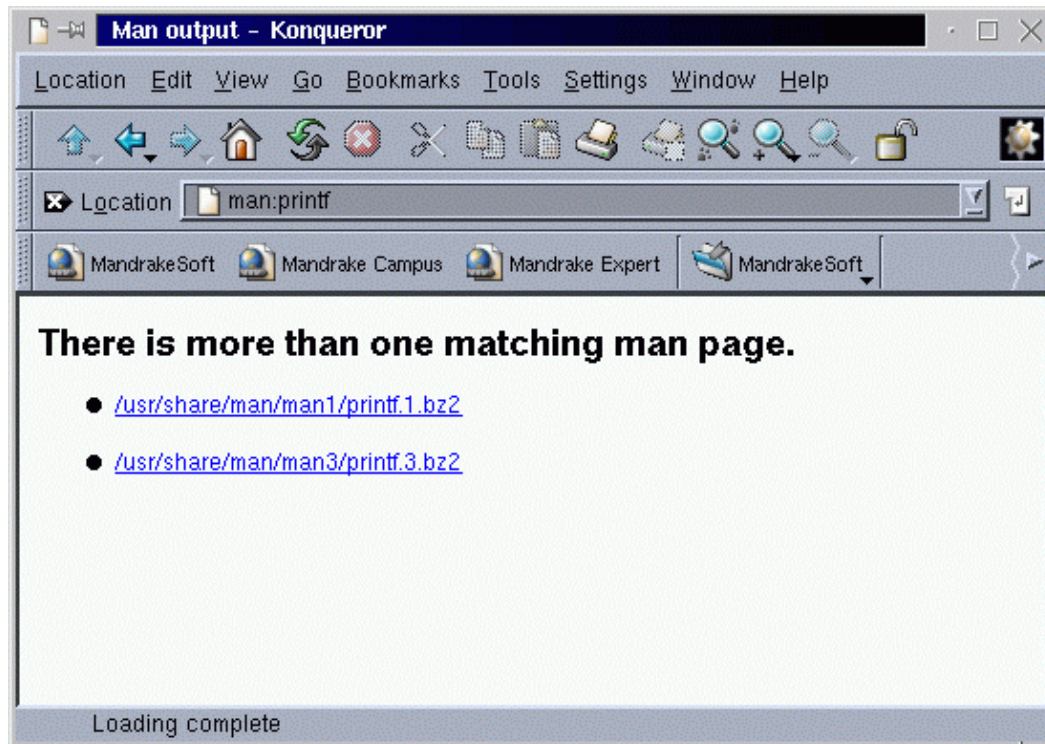
Las diferentes secciones son:

Sección

- 1 Comandos de usuario.
- 2 Llamadas al sistema, es decir, funciones provistas por el kernel.
- 3 Subrutinas, es decir, funciones de biblioteca.
- 4 Dispositivos, es decir, ficheros especiales en el directorio /dev.
- 5 Descripciones de formato de ficheros, e.g. /etc/passwd.
- 6 Juegos, auto-explicativo.
- 7 Miscelánea, e.g. paquetes de macro, convenciones.
- 8 Herramientas de administración del sistema que sólo puede ejecutar el superusuario.
- 9 Otros.
- n Nueva documentación, que debería cambiarse a una sección más apropiada.
- l Documentación local sobre este sistema en particular.

De esta forma con "man 1 printf" obtenemos la documentación sobre el comando de shell printf y con "man 3 printf" se mostrará la descripción de la función de biblioteca C. Ejecutando exactamente "man printf" se imprimirá la página que primero se encuentre (normalmente printf de la sección 1).

Para comprobar si existen varias versiones de páginas de manual se puede utilizar el comando wichman como se ha mostrado anteriormente (descargable desde [mi página](#), o bien tecleando simplemente "man:printf" en konqueror que nos devolverá:



MANPATH

El comando `man` busca las páginas de manual según el valor de la variable de entorno `MANPATH`. Desgraciadamente muchas distribuciones Linux establecen esta variable de forma incorrecta. A menudo no se incluye `/usr/lib/perl5/man`, que contiene un valioso conjunto de documentación sobre todas las funciones perl. Puede añadirlo a su `MANPATH` (en `.bashrc` o `.tcshrc` o ...) así:

Bash:

```
MANPATH="/usr/local/man:/usr/man:/usr/share/man:/usr/X11R6/man:/usr/lib/perl5/man"
export MANPATH
```

Tcsh:

```
setenv MANPATH "/usr/local/man:/usr/man:/usr/share/man:/usr/X11R6/man:/usr/lib/perl5/man"
```

Después de definir la variable `MANPATH` se puede probar "`man Pod::Man`" para ver si se accede a las páginas de Perl.

Macros de formato

Escribir una página de manual es muy fácil. Es un sencillo lenguaje de composición donde las palabras clave del lenguaje de composición empiezan con un punto en el inicio de la línea. Estas palabras clave también se llaman macros. Las macros más importantes son:

```
.TH -> Inicia el título/cabecera de la página de manual
.SH -> Encabezado de sección
.PP -> Nuevo párrafo
." -> Comentario
.TP -> Sangrado del texto que esté 2 líneas por debajo de esta macro
```

La sintaxis de .TH es:

.TH [nombre del programa] [número de sección] [centro pie de página] [izquierda pie de página] [centro encabezado]

La sintaxis .SH es:

.SH encabezamiento

La sintaxis de .PP es muy sencilla. Simplemente produce un avance de línea.

He visto que a veces es útil incluir texto preformateado para ejemplos de código de programas. Esto se puede hacer con:

```
.nf
_el_texto_pre_formateado_
_va_aquí_____
.fi
```

Tengamos en cuenta que éstas son macros groff/nroff y debido a esto no pertenecen a las macros especiales de páginas de manual. Sin embargo, deberían funcionar bien en todos los sistemas Unix.

Todas las macros para formatear páginas de manual están documentadas en la página de manual llamada `groff_man(7)` ([Aquí se puede ver una versión en html de la página de manual groff_man\(7\)](#)). No voy a explicar estas macros, sino que sugiero que se lea la página de manual de `groff_man`. La página `groff_man` está muy detallada y contiene todo lo que se necesita saber.

Los capítulos

Antes de empezar a escribir nuestra propia página deberíamos saber que las páginas de manual normalmente se estructuran en capítulos. Por convención los posibles encabezamientos de los capítulos son los siguientes:

NOMBRE	Nombre de sección, el nombre de la función o el comando.
SINOPSIS	Uso.
DESCRIPCION	Descripción general.
OPCIONES	Descripciones y parámetros.
VALOR DEVUELTO	Secciones dos y tres llamadas a funciones.
ENTORNO	Describe variables de entorno.
FICHEROS	Ficheros asociados.
EJEMPLOS	Ejemplos y consejos.
DIAGNOSTICOS	Normalmente usado por la sección 4 diagnósticos de dispositivos e interfaces.
ERRORES	Secciones dos y tres errores y señales de manejadores.
VEASE TAMBIEN	Referencias cruzadas y citas.
CONFORME A	Conformidad con estándares, en su caso.
FALLOS	Errores y advertencias.
SEGURIDAD	Aspectos de seguridad a tener en cuenta.
otros	Se pueden añadir cabeceras hechas a medida por los autores.

(N.T. en inglés estos encabezamientos son respectivamente NAME, SYNOPSIS, DESCRIPTION, OPTIONS, RETURN VALUES, ENVIRONMENT, FILES, EXAMPLES, DIAGNOSTICS, ERRORS, SEE ALSO, STANDARDS, BUGS, SECURITY CONSIDERATIONS)

Una página de manual de ejemplo

Aquí tenemos una página de manual de ejemplo. Hay que tener en cuenta que se necesita \- para distinguir la raya de los guiones. Teclee todo esto en su editor de texto y guárdelo como cdspeed.1.

```
.TH cdspeed 1 "10 de Septiembre de 2003" "version 0.3" "COMANDOS DE USUARIO"
.SH NOMBRE
cdspeed \- reduce la velocidad del cdrom para obtener un tiempo de
acceso más rápido
.SH SINOPSIS
.B cdspeed
[\-h] [\-d dispositivo] \-s velocidad
.SH DESCRIPCION
Las unidades de cdrom modernas son demasiado rápidas. Puede
llevar bastantes segundos en una unidad de cdrom de velocidad 60x
empezar a girar y leer datos de la unidad. El resultado en estas
unidades es precisamente mucho más lento que en unidades a 8x o 24x.
Esto es particularmente cierto si sólo se lee ocasionalmente (e.g.
cada 5 segundos) ficheros pequeños. Esta utilidad limita la velocidad
y hace que la unidad responda mejor cuando se accede a ficheros pequeños.
.PP
cdspeed también hace a la unidad menos ruidosa lo que es muy
útil para escuchar música en el ordenador.
.SH OPCIONES
.TP
\-h
presenta un pequeño texto de ayuda
.TP
\-d
usa el dispositivo dado en lugar de /dev/cdrom
.TP
\-s
establece la velocidad. El argumento es un entero. Cero significa restaurar
a la velocidad máxima
.SH EJEMPLOS
.TP
Establecer la velocidad máxima del cdrom a 8:
.B cdspeed
\-s 8
.TP
Restaurar la velocidad máxima:
.B cdspeed
\-s 0
.PP
.SH ESTADO DE SALIDA
cdspeed devuelve un cero si consigue cambiar la velocidad máxima de la
unidad de cdrom a la establecida. Se devuelve no cero en caso de fallo.
.SH AUTOR
Guido Socher (guido (at) linuxfocus.org)
.SH VEASE TAMBIEN
eject(1)
```

Pulse [aquí \(cdspeed.html\)](#) para ver la página anterior (N.T. en su versión inglesa).

Viendo y formateando nuestra página de manual

Mientras escribimos la página de manual podemos ir viendo de vez en cuando qué aspecto tiene. Tecleando:

```
nroff -man nuestra_página_de_manual.1 | less
```

o

```
groff -man -Tascii nuestra_página_de_manual.1 | less
```

Para convertir una página de manual en texto plano preformateado (e.g. para comprobar la ortografía) se utiliza:

```
nroff -man nuestra_página_de_manual.1 | col -b > xxxx.txt
```

Para convertirlo en Postscript (para imprimir o para una conversión posterior a pdf) se usa:

```
groff -man -Tps nuestra_página_de_manual.1 > nuestra_página_de_manual.ps
```

Se puede convertir la página de manual a html mediante:

```
man2html nuestra_página_de_manual.1
```

Usando perl POD para generar páginas de manual

Sé que a muchos les estará extrañando el hecho de sólo poder editar la página de manual en un editor de texto. Lo que quieren es generar la página de manual. El formato de documentación POD de perl es una buena elección. Podemos escribir la página en sintaxis POD y entonces ejecutar el comando

```
pod2man nuestra_página_de_manual.pod > nuestra_página_de_manual.1
```

La sintaxis del lenguaje de documentación pod de perl se describe en una página de manual llamada perlpod. La página de manual del ejemplo anterior se podría ver en formato pod como se muestra a continuación. Hay que tener en cuenta que POD es sensitivo a los espacios por lo que las líneas en blanco alrededor de "=head" son necesarias.

```
=head1 NOMBRE
```

```
cdspeed - reduce la velocidad del cdrom para obtener un tiempo de  
          acceso más rápido
```

```
=head1 SINOPSIS
```

```
cdspeed [-h] [-d dispositivo] -s velocidad
```

```
=head1 DESCRIPCION
```

```
Las unidades de cdrom modernas son demasiado rápidas. Puede  
llevar bastantes segundos en una unidad de cdrom de velocidad 60x  
empezar a girar y leer datos de la unidad. El resultado en estas  
unidades es precisamente mucho más lento que en unidades a 8x o 24x.  
Esto es particularmente cierto si sólo se lee ocasionalmente (e.g.  
cada 5 segundos) ficheros pequeños. Esta utilidad limita la velocidad  
y hace que la unidad responda mejor cuando se accede a ficheros pequeños.
```

```
cdspeed también hace a la unidad menos ruidosa lo que es muy  
útil para escuchar música en el ordenador.
```

```
=head1 OPCIONES
```

```
B<-h> presenta un pequeño texto de ayuda
```

B<-d> usa el dispositivo dado en lugar de /dev/cdrom

B<-s> establece la velocidad. El argumento es un entero. Cero significa restaurar a la velocidad máxima

=head1 EJEMPLOS

Establecer la velocidad máxima del cdrom a 8:

```
cdspeed -s 8
```

Restaurar la velocidad máxima:

```
cdspeed -s 0
```

=head1 ESTADO DE SALIDA

cdspeed devuelve un cero si consigue cambiar la velocidad máxima de la unidad de cdrom a la establecida. Se devuelve no cero en caso de fallo.

=head1 AUTOR

Guido Socher

=head1 VEASE TAMBIEN

eject(1)

Referencias

- Man-page HOWTO
- groff man(7), man-page macros

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Guido Socher "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Guido Socher (homepage) en --> es: Roberto Hernando Velasco (homepage)</p>
--	--

2005-01-10, generated by lfparsr_pdf version 2.51