

Página Man ¿cómo crearlas?

Si alguna vez creaste al menos una herramienta para GNU/Linux, muy probablemente se te haya cruzado la pregunta «¿Cómo se hará para crear una página Man?» y la idea de este artículo, es darte una respuesta paso a paso.

Escrito por: **Eugenia Bahit** (GLAMP Hacker & eXtreme Programmer)



Eugenia es **Arquitecta de Software**, docente e instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y **eXtreme Programming**. Miembro de la **Free Software Foundation**, **The Linux Foundation** y **Debian Hackers**. Creadora de **python-printr**, **Europio Engine** y colaboradora de **Vim**.

Webs:

Cursos de programación: www.cursosdeprogramacionadistancia.com
Web personal: www.eugeniabahit.com

Redes sociales:

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

Man es la herramienta utilizada por el sistema operativo GNU con Linux -así como por otros basados en *nix-, para documentar desde comandos hasta aplicaciones. Generalmente, como usuarios del sistema operativo GNU, cuando necesitamos aprender sobre un determinado comando ejecutamos `man comando` para acceder a la **página del manual** específica para el comando correspondiente.

Crear nuevos comandos es tan simple como crear un script ejecutable y almacenarlo en el directorio `/sbin` (o `/usr/sbin`), sin embargo, crear una página del manual (*man page*) para dicho comando, muchas veces suele ser “una receta misteriosa”, tal vez, por la escasa -o difícilmente accesible- información al respecto.

Siguiendo la filosofía de “erradicación de misterios” que nos acompaña edición tras edición, en esta oportunidad, aprenderemos a crear nuestras propias *man pages*.

Aclarando el panorama

Las páginas del manual son archivos de texto plano que sorprendentemente, pueden ser escritos con cualquier editor de texto, sin requerir de herramientas específicas para su creación (muy a pesar de los tutoriales que afirman lo contrario).

Una forma de “descubrir” por tus propios medios de qué se trata una página del manual, es localizar cualquier *man page* de un comando al azar. Por ejemplo, para localizar la ubicación del archivo de la página del manual para el comando `mkdir`, podemos ayudarnos del comando `whereis`:

```
whereis mkdir
```

Este comando, no solo nos arrojará la ruta del binario, sino además, la de la página del manual:

```
/usr/share/man/man2/mkdir.2.gz
```

Para visualizar el archivo, solo bastará con copiarlo y descomprimirlo con `gzip`:

```
# cp /usr/share/man/man2/mkdir.2.gz /home/user/  
# gzip -d mkdir.2.gz
```

Si miramos el contenido del archivo descomprimido, podremos ver que se trata de texto plano puro:

```
(...)  
The newly created directory will be owned by the effective user ID of the  
process.  
If the directory containing the file has the set-group-ID  
bit set, or if the file system is mounted with BSD group semantics  
.RI ( "mount -o bsdgroups"  
or, synonymously  
.IR "mount -o grpuid" ),  
the new directory will inherit the group ownership from its parent;  
otherwise it will be owned by the effective group ID of the process.  
  
If the parent directory has the set-group-ID bit set then so will the  
newly created directory.  
.SH "RETURN VALUE"  
.BR mkdir ()  
returns zero on success, or -1 if an error occurred (in which case,  
.I errno  
is set appropriately).  
.SH ERRORS  
.TP  
.B EACCESS  
The parent directory does not allow write permission to the process,  
or one of the directories in  
.I pathname  
did not allow search permission.  
(See also  
.BR path_resolution (7).)  
.TP  
.B EEXIST  
.I pathname
```

(...)

Identificar el número de sección

Si observaste detenidamente la salida del `whereis` que hicimos anteriormente, habrás notado que tanto la carpeta donde se almacena la página del manual, como la página del manual propiamente dicha, tienen un número:

```
/usr/share/man/man2/mkdir.2.gz
```

El número en cuestión, varía dependiendo de la herramienta o comando. Por ejemplo, en el caso de `iptables`, el número correspondiente es el 8:

```
$ whereis iptables
iptables: /sbin/iptables /usr/share/iptables /usr/share/man/man8/iptables.8.gz
```

Estos números, representan la **sección del manual** a la cuál pertenece el comando o herramienta. El manual se encuentra dividido en 8 secciones a las cuáles a cada una le corresponde un número:

```
1 Programas ejecutables y guiones del intérprete de órdenes
2 Funciones provistas por el núcleo (kernel) del Sistema Operativo
3 Funciones de la biblioteca del propio sistema
4 Archivos especiales (se encuentran generalmente en /dev)
5 Formato de ficheros y convenios p.ej. I/etc/passwd
6 Juegos
7 Paquetes de macros y convenios p.ej. man(7), groff(7).
8 Órdenes de administración del sistema (generalmente solo son para root)
```

Por lo tanto, para crear una página del manual, lo primero que debemos hacer es identificar la sección a la cuál pertenece el comando o herramienta que estamos a punto de documentar.

El número de sección nos servirá para:

1. Saber **en qué directorio** dentro de `/usr/share/man/` tendremos que guardar nuestra propia página;
2. Establecer el **nombre de archivo**, ya que el número de sección, formará parte del nombre del archivo.

Estableciendo un nombre

Si haces un listado de cualquiera de las carpetas dentro de `/usr/share/man/` o ejecutas

el comando `whereis` para varios comandos, podrás notar que el nombre de los archivos de las páginas del manual, siempre guardan el mismo formato:

```
comando.numero-seccion
```

Por ejemplo, para el comando `adduser` el archivo de la página del manual se llama `adduser.8` mientras que para el comando `cp`, se llama `cp.1`.

Contenido del archivo

Cuando hicimos un `cat` del archivo `mkdir.2`, como bien pudo observarse, algunas líneas comenzaban por un punto seguido de una o dos letras mayúsculas:

```
.I errno
is set appropriately).
.SH ERRORS
.TP
.B EACCES
The parent directory does not allow write permission to the process,
or one of the directories in
.I pathname
```

Las mismas, se denominan **macros de formato** y forman parte de los sistemas de formato de texto para plataformas tipo `*nix`, como es el caso del sistema operativo GNU, el cuál utiliza a Linux como Kernel.

roff fue el programa original que permitía dar formato al texto. Luego, fueron apareciendo programas similares como nroff y troff (entre otros), hasta que en 1990, James Clark creó groff, la alternativa libre del proyecto GNU

Las **macros de formato** pueden consultarse accediendo a la página del manual `man` de la sección 7, mediante la ejecución de: `man 7 man`

Título

La línea de título debe ser la primera luego de los comentarios sobre el archivo que eventualmente pudiesen colocarse. Un título es indicado con la macro `.TH` y debe indicarse con la siguiente sintaxis:

```
.TH nombre-comando numero-sección fecha fuente-origen título-del-manual
```

Por ejemplo, la siguiente línea:

```
.TH blockip 8 2013-06-29 "JackTheStripper 1.0 beta 3" "Manual de blockip"
```

Producirá:

blockip(8)	Manual de blockip	blockip(8)
JackTheStripper 1.0 beta 3	2013-06-29	blockip(8)

Secciones

Las secciones se indican con la macro `.SH` seguida del nombre de sección:

```
.SH nombre de la sección
```

Entre las secciones de uso más frecuente que podremos indicar, encontramos las siguientes:

NAME /NOMBRE

Nombre del comando, archivo o herramienta

SYNOPSIS /SINOPSIS

Sintaxis de uso

DESCRIPTION /DESCRIPCION

Descripción del comando, archivo o herramienta

OPTIONS/OPCIONES

Descripción de las opciones aceptadas en la línea de comandos por el comando o herramienta.

Normalmente `splo` se implementa en las secciones 1 y 8.

ENVIRONMENT /ENTORNO

Muestra una lista de las variables de entorno utilizadas por el comando o herramienta, describiendo cómo éstas son utilizadas

FILES /ARCHIVOS

Una lista de todos los archivos utilizados por el comando o herramienta.

BUGS /ERRORES

Descripción de errores o problemas conocidos.

EXAMPLE /EJEMPLO

Ejemplos de uso del comando o herramienta

AUTHORS/AUTORES

Lista de los nombres de los autores.
Generalmente se utiliza el formato: Nombre del Autor <email@example.org>

SEE ALSO/VÉASE TAMBIÉN

Una lista de páginas man sugeridas, separadas por coma y ordenadas primero, por número de sección y luego, por orden alfabético

Veamos un pequeño ejemplo y su salida:

```
.TH blockip 8 2013-06-29 "JackTheStripper 1.0 beta 3" "Manual de blockip"  
.SH NOMBRE  
blockip – bloquea una IP de forma permanente
```

El archivo anterior, produciría una salida como la siguiente:

```
blockip(8)                Manual de blockip                blockip(8)  
NOMBRE  
    blockip – bloquea una IP de forma permanente  
JackTheStripper 1.0 beta 3    2013-06-29                blockip(8)
```

Una completa descripción de las posibles **secciones** que pueden implementarse en una página del manual, podemos obtenerla consultando la página man-pages de la sección 7, ejecutando: **man 7 man-pages**

Formato de fuente

Entre los formatos más usados podemos encontrar:

```
.B Negrita  
.B hola mundo  
produce: hola mundo  
  
.I Itálica  
.I hola mundo  
produce: hola mundo  
  
.R Romana  
.R hola mundo  
produce: hola mundo
```

Los formatos anteriores pueden además, utilizarse en cualquier combinación, produciendo así un alternado de formato. Veamos algunos ejemplos:

```
.BI negrita alternado itálica
    negritaalternadoitálica

.IB itálica alternado negrita
    itálicaalternadonegrita

.BR negrita alternando Romana
    negritaalternandoRomana

.RI romana alternando itálica
    romanaalternandoitálica
```

Párrafos y saltos de línea

Un nuevo **párrafo** se indica con la macro **.PP** mientras que un **salto de línea**, con **.br**

```
.TH micomando 8 2013-06-29 "MiHerramienta 1.0" "Manual de micomando"
.SH NOMBRE
micomando – ejemplo para página del manual
.SH DESCRIPCIÓN
.B micomando
lo escribí en negritas y fui a la siguiente línea para finalizar las negritas
.PP
Ahora escribo un nuevo párrafo
.br
que tiene un salto de línea
```

La salida del ejemplo anterior, será:

```
micomando(8)                Manual de micomando                micomando(8)

NOMBRE
    micomando – ejemplo para página del manual

DESCRIPCIÓN
    micomando lo escribí en negritas y fui a la siguiente línea para
    finalizar las negritas

    Ahora escribo un nuevo párrafo
    que tiene un salto de línea

MiHerramienta 1.0          2013-06-29          micomando(8)
```

Con **.TP** es posible indicar un párrafo con **columnas**, ideal para listar argumentos:

```
.TH micomando 8 2013-06-29 "MiHerramienta 1.0" "Manual de micomando"
.SH NOMBRE
micomando – ejemplo para página del manual
.SH DESCRIPCIÓN
.B micomando
lo escribí en negritas y fui a la siguiente línea para finalizar las negritas
.PP
```

```
Ahora escribo un nuevo párrafo
.br
que tiene un salto de línea
.SH OPCIONES
.TP
.B -a
describe lo que hace la opción -a
.TP
.B -b
describe lo que hace la opción -b
```

Lo anterior, se verá así:

```
micomando(8)                Manual de micomando                micomando(8)
NOMBRE
    micomando – ejemplo para página del manual
DESCRIPCIÓN
    micomando lo escribí en negritas y fui a la siguiente línea para
    finalizar las negritas
    Ahora escribo un nuevo párrafo
    que tiene un salto de línea
OPCIONES
    -a    describe lo que hace la opción -a
    -b    describe lo que hace la opción -b
MiHerramienta 1.0           2013-06-29           micomando(8)
```

Cuando se observa todo lo que hasta ahora ha sido probado, se puede notar que los saltos de línea no son respetados a no ser que se implemente la macro `.br`. Sin embargo, esto puede ser muy tedioso si lo que se desea es, por ejemplo, indicar un fragmento de código fuente.

Para respetar los saltos de línea (e incluso los espacios y tabulaciones), disponemos de las macros `.nf` y `.fi`, para dar inicio y fin a un **bloque de texto con pre-formato**:

```
.nf
for elemento in tupla:
    print elemento * 2
.fi
```

Con las macros `.nf` y `.fi`, nuestro texto con que pre-formato, ha sido respetado:

```
for elemento in tupla:
    print elemento * 2
```


Comentarios

Los comentarios del archivo en general, deberán indicarse al comienzo del archivo. Sin embargo, comentarios sobre el propio texto, pueden hacerse en cualquier lugar que sea necesario.

Para escribir comentarios se utiliza la macro `.\`

Un ejemplo de comentarios, es el que vimos a la hora de hacer un `cat` a la página del manual para `mkdir` en la sección 2:

```
.\" Hey Emacs! This file is -*- nroff -*- source.
.\"
.\" This manpage is Copyright (C) 1992 Drew Eckhardt;
.\"                               1993 Michael Haardt
.\"                               1993,1994 Ian Jackson.
.\" You may distribute it under the terms of the GNU General
.\" Public License. It comes with NO WARRANTY.
.\"
```

Implementando nuestra nueva página `man`

Finalmente, cuando nuestro archivo está terminado solo restará comprimirlo con `gzip`:

```
gzip -q /usr/share/man/man8/micomando.8
```

Por favor, notar que la opción `-q` del comando `gzip`, reemplazará el archivo `micomando.8` por `micomando.8.gz`.

Como puedes ver, las opciones de formato de `groff` son inmensas y aquí, solo hemos mencionado una mínima parte. Una buena forma de entender más a fondo tanto las macros de formato como la estructura interna de una página de manual, es mirando el fuente de los archivos, compararlo con la página del manual e ir tratando de implementar formatos similares y viendo el reflejado el resultado visual del mismo.

Y ahora ¡a documentar se ha dicho!