



par Guido Socher (homepage)

L'auteur:

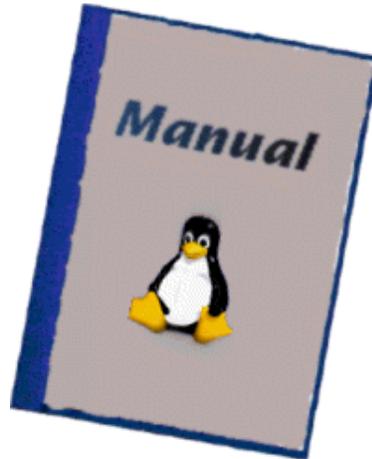
Guido aime Linux parce que c'est un système d'exploitation très souple, qui offre beaucoup plus de possibilités que n'importe quel autre.

Traduit en Français par:

Christophe Bénard

<Christophe.BENARD(at)wanadoo.fr>

Ecrire des pages de manuel



Résumé:

Tout bon programme utilisable depuis le shell d'Unix devrait être documenté avec ses propres pages de manuel (man-pages). Ce tutoriel constitue une introduction rapide à l'écriture de ces pages.

Introduction

La documentation compte souvent davantage que le logiciel lui-même, particulièrement si ce logiciel doit être utilisé par d'autres personnes que l'auteur. Même si j'écris un programme que je n'envisage pas de publier, je rédige une documentation car quelques mois plus tard, je pourrais avoir oublié comment l'utiliser, et une documentation bien structurée m'indiquera en quelques secondes comment en faire usage.

Les utilitaires traditionnels utilisables en ligne de commandes sous Linux ont toujours été documentés sous la forme de pages de manuel. Un simple `man nom_de_commande` vous explique comment utiliser la commande.

L'avantage des pages de manuel sur toutes les autres formes de documentation est que

1. Elles peuvent être affichées en quelques secondes sur n'importe quel terminal Linux
2. Elles peuvent être facilement converties en d'autres formats: HTML, PDF, Postscript, Text,...

3. Les pages de manuel peuvent être lues, non seulement dans une fenêtre de terminal, mais aussi avec d'autres programmes tels que `konqueror` (tapez juste: `man:nom_de_commande`)

Les sections

Les pages de manuel sont structurées en sections, tout comme un livre est structuré en chapitres. Il existe par exemple deux pages de manuel sur `printf`. L'une concernant la fonction de la librairie C (section 3) et l'autre à propos de la commande de shell `printf` (section 1):

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

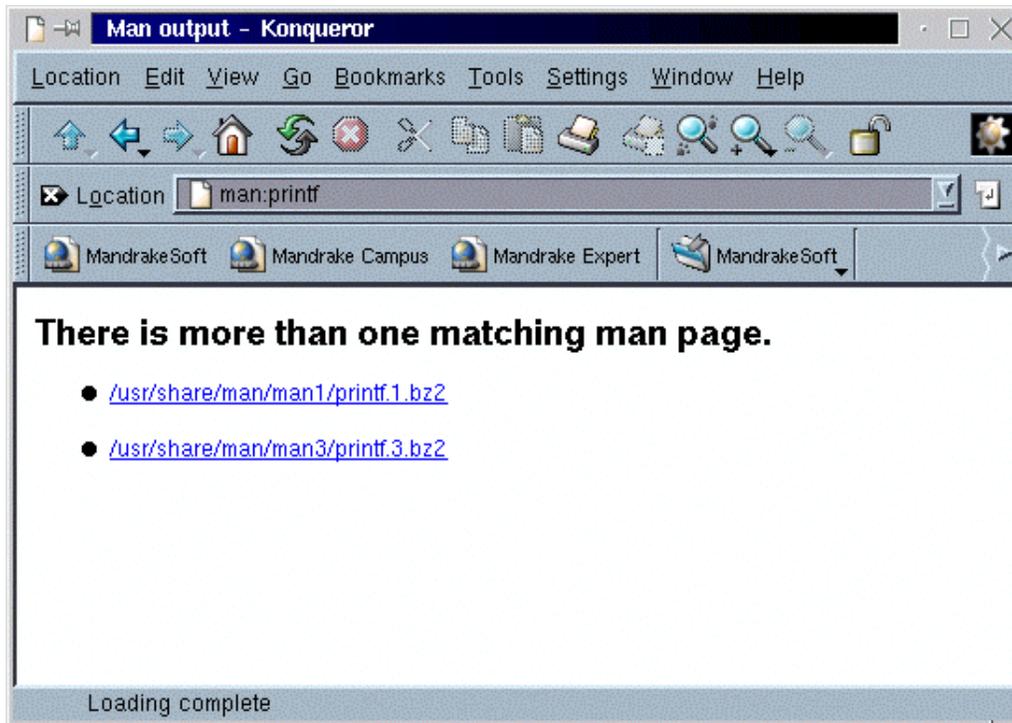
Les différentes sections sont les suivantes:

Section

- 1 Commandes du niveau de l'utilisateur
- 2 Appels système, c'est à dire, fonctions offertes par le noyau.
- 3 Sous-routines, c'est à dire, fonctions de librairie.
- 4 Périphériques, c'est à dire, fichiers spéciaux comme les entrées de `/dev`.
- 5 Descriptions de format de fichiers, par exemple `/etc/passwd`.
- 6 Jeux, auto-documentation.
- 7 Divers, par exemple package de langage, conventions.
- 8 Outils d'administration que seul `root` peut exécuter.
- 9 Autre
- n Documentations récentes, qui pourraient être déplacées vers une section plus appropriée.
- l Documentation locale faisant référence à ce système en particulier.

Par conséquent, taper "`man 1 printf`" affichera la documentation relative à la commande de shell `printf` tandis que "`man 3 printf`" affichera la description de la fonction de la librairie C. En limitant la saisie à "`man printf`" on affichera la première page trouvée (en général, la section 1 de `printf`).

Pour vérifier s'il existe plusieurs versions de pages de manuel, on peut utiliser la commande `whichman`, comme indiqué ci-dessus (téléchargement depuis ma page perso), ou juste taper "`man:printf`" dans `konqueror`, ce qui vous indiquera:



MANPATH

La commande man cherche les pages de manuel en se basant sur la valeur de la variable d'environnement MANPATH. Malheureusement dans de nombreuses distributions Linux, ce chemin est initialisé de façon incorrecte. Souvent /usr/lib/perl5/man, qui permet l'accès à une documentation abondante sur toutes les fonctions perl, n'est pas inclus. Vous pouvez l'ajouter à MANPATH (dans le fichier .bashrc ou .tcshrc ou ...) de cette manière:

```
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/per
```

Après avoir initialisé le chemin d'accès aux pages de manuel, vous pouvez essayer "man Pod::Man" pour voir si vous obtenez l'une des pages concernant perl.

Les mots-clé de formatage

Ecrire une page de manuel est très facile. Il s'agit d'un langage simple dont les mots clés utilisés comme balises commencent par un point en début de chaque ligne. Ces mots clés sont aussi appelés macros. Les macros les plus importantes sont:

```
.TH -> introduit le titre/l'entête de la page de manuel
.SH -> Section entête
```

```
.PP -> Nouveau paragraphe
." -> Ligne de commentaire
.TP -> Indente le texte qui vient 2 lignes après cette macro
```

La syntaxe de .TH est:

```
.TH [nom du programme] [numéro de section] [centre du pied de page] [section gauche du pied de page]
[section centrale de l'entête]
```

La syntaxe de .SH est:

```
.SH texte du titre
```

La syntaxe de .PP est très simple. Elle produit juste un saut de ligne.

Je trouve parfois pratique d'insérer du texte pré-formaté pour afficher des exemples de codes de programmes. Voici comment procéder:

```
.nf
_votre_texte_pré-formaté_
_ici_____
.fi
```

Notez que ce sont des macros groff/nroff et donc, qu'elles n'appartiennent pas aux macros spécifiques à la rédaction de pages de manuel. Il semblerait cependant qu'elles fonctionnent correctement sur tous les systèmes Unix.

Toutes les macros de formatage de page de manuel sont documentées dans la page de manuel appelée `groff_man(7)` (Cliquez [ici](#) pour voir la version html de la page de manuel de `groff_man(7)`). Je ne vais pas expliquer ici l'utilisation des macros, mais je vous suggère plutôt de lire la page `groff_man`. La page `groff_man` est très détaillée et contient tout ce que vous avez besoin de savoir.

Les chapitres

Avant de commencer à écrire votre propre page de manuel, vous devez savoir que les pages de manuel sont normalement structurées en chapitres. Voici la liste des titres de chapitres que, par convention, vous pouvez utiliser:

| | |
|---------------|---|
| NAME | Section concernant le nom de la fonction ou de la commande. |
| SYNOPSIS | Usage. |
| DESCRIPTION | Description Général |
| OPTIONS | Doit indiquer les options et paramètres. |
| RETURN VALUES | Appels de fonction des sections deux et trois. |
| ENVIRONMENT | Décrit les variables d'environnement. |
| FILES | Fichiers associés au sujet. |
| EXAMPLES | Exemples et suggestions. |
| DIAGNOSTICS | Normalement utilisé pour la section 4 diagnostique des interfaces de périphériques. |
| ERRORS | Sections deux et trois gestion d'erreur et de signal. |

SEE ALSO Références croisées et citations.
STANDARDS Conformité aux standards, si c'est applicable.
BUGS Pièges et avertissements.
SECURITY CONSIDERATIONS
 Considérations liée à la
 sécurité dont il faut être informé.
other Des entêtes personnalisés peuvent être
 ajoutés, à la discretion de l'auteur.

Un exemple de page de manuel

Voici un petit exemple de page de manuel. Il convient de noter que le \- est requis pour distinguer le tiret des traits d'union. Saisissez tout ce qui suit grâce avec votre éditeur de texte, et enregistrez le document sous le nom cdspeed.1.

```
.TH cdspeed 1 "September 10, 2003" "version 0.3" "USER COMMANDS"
.SH NAME
cdspeed \- decrease the speed of you cdrom to get faster access time
.SH SYNOPSIS
.B cdspeed
[\-h] [\-d device] \-s speed
.SH DESCRIPTION
Modern cdrom drives are too fast. It can take several seconds
on a 60x speed cdrom drive to spin it up and read data from
the drive. The result is that these drives are just a lot slower
than a 8x or 24x drive. This is especially true if you are only
occasionally (e.g every 5 seconds) reading a small file. This
utility limits the speed and makes the drive more responsive
when accessing small files.
.PP
cdspeed makes the drive also less noisy and is very useful if
you want to listen to music on your computer.
.SH OPTIONS
.TP
\-h
display a short help text
.TP
\-d
use the given device instead of /dev/cdrom
.TP
\-s
set the speed. The argument is a integer. Zero means restore maximum
speed.
.SH EXAMPLES
.TP
Set the maximum speed to 8 speed cdrom:
.B cdspeed
\-s 8
.PP
.TP
Restore maximum speed:
.B cdspeed
\-s 0
.PP
.SH EXIT STATUS
cdspeed returns a zero exist status if it succeeds to change to set the
```

```
maximum speed of the cdrom drive. Non zero is returned in case of failure.  
.SH AUTHOR  
Guido Socher (guido (at) linuxfocus.org)  
.SH SEE ALSO  
eject(1)
```

Cliquez [ici \(cdspeed.html\)](#) pour visualiser la page décrite ci-dessus.

Visualiser et formater votre page de manuel

Pendant que vous écrivez votre page de manuel, vous pouvez, de temps à autre, la visualiser pour vérifier qu'elle est correcte. Tapez:

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

ou

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

Pour convertir une page de manuel en texte plat pré-formaté (pour contrôler l'orthographe par exemple), utilisez la commande:

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

Pour la convertir au format Postscript (pour l'imprimer ou la convertir au format pdf), utilisez la commande:

```
groff -man -Tps votre_page_de_manuel.1 > your_manpagefile.ps
```

Pour convertir la page de manuel au format html, utilisez la commande:

```
man2html votre_page_de_manuel.1
```

Utiliser perl POD pour générer des pages de manuel

Je sais que nombreux sont ceux qui trouvent étrange de concevoir une page de manuel avec un éditeur de texte. Ils préfèrent générer la page de manuel. Le format de documentation perl POD constitue un bon choix. Il est possible d'écrire la page en utilisant la syntaxe POD puis lancer la commande

```
pod2man votre_page_de_manuel.pod > votre_page_de_manuel.1
```

La syntaxe du langage de documentation perl pod est décrit dans une page de manuel intitulée perlpod. L'exemple de page de manuel ci-dessus ressemblerait à ce qui suit dans le format pod. Il faut noter que POD est sensible aux espaces et que les lignes vides autour de "=head" sont également nécessaires.

```
=head1 NAME
```

```
cdspeed - decrease the speed of you cdrom to get faster access time
```

```
=head1 SYNOPSIS
```

```
cdspeed [-h] [-d device] -s speed
```

=head1 DESCRIPTION

Modern cdrom drives are too fast. It can take several seconds on a 60x speed cdrom drive to spin it up and read data from the drive. The result is that these drives are just a lot slower than a 8x or 24x drive. This is especially true if you are only occasionally (e.g every 5 seconds) reading a small file. This utility limits the speed and makes the drive more responsive when accessing small files.

cdspeed makes the drive also less noisy and is very useful if you want to listen to music on your computer.

=head1 OPTIONS

B<-h> display a short help text

B<-d> use the given device instead of /dev/cdrom

B<-s> set the speed. The argument is a integer. Zero means restore maximum speed.

=head1 EXAMPLES

Set the maximum speed to 8 speed cdrom:

```
cdspeed -s 8
```

Restore maximum speed:

```
cdspeed -s 0
```

=head1 EXIT STATUS

cdspeed returns a zero exist status if it succeeds to change to set the maximum speed of the cdrom drive. Non zero is returned in case of failure.

=head1 AUTHOR

Guido Socher

=head1 SEE ALSO

eject(1)

Références

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

Site Web maintenu par l'équipe d'édition

LinuxFocus

© Guido Socher

"some rights reserved" see

linuxfocus.org/license/

<http://www.LinuxFocus.org>

Translation information:

en --> -- : Guido Socher (homepage)

en --> fr: Christophe Bénard

<Christophe.BENARD(at)wanadoo.fr>