



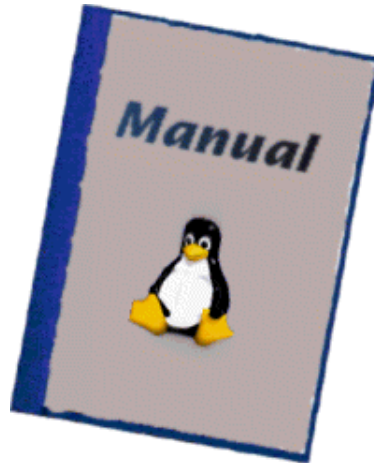
door Guido Socher (homepage)

Over de auteur:

Guido houdt van Linux omdat het erg flexibel is en veel meer mogelijkheden biedt dan enig ander besturingssysteem.

Vertaald naar het Nederlands door:
Guus Snijders
<ghs(at)linuxfocus.org>

man-pagina's schrijven



Kort:

Ieder goed programma dat gebruikt kan worden vanuit de Unix shell zou gedocumenteerd moeten zijn in zijn eigen man-page. Deze tutorial geeft een korte introductie tot het schrijven van manual pages.

Introductie

Documentatie is vaak belangrijker dan de software zelf, vooral als de software niet meer door de auteur gebruikt wordt. Zelfs als ik een programma schrijf dat niet bedoeld is om te publiceren, schrijf ik er documentatie voor omdat ik een paar maanden later vergeten ben hoe het programma gebruikt moet worden en goed gestructureerde documentatie vertelt me in enkele seconden hoe het programma zich laat bedienen.

Traditionele Linux commando regelutilities werden (en worden) altijd gedocumenteerd in man-pages. Een eenvoudige `man commandonaam` vertelt je hoe het programma te gebruiken.

Het voordeel van man-pages ten opzichte van andere vormen van documentatie is dat

1. Ze kunnen in seconden bekeken worden in een Linux terminal
2. Ze kunnen eenvoudig worden geconverteerd naar andere formaten: HTML, PDF, Postscript, Tekst, ...

3. Man pages kunnen niet alleen bekeken worden in terminal vensters maar ook in andere programma's zoals konqueror (type: man:commandonaam)

De secties

Man-pages zijn opgedeeld in secties net zoals een boek is gestructureerd in hoofdstukken. Er zijn bijvoorbeeld twee man-pages over printf. Een voor de C-library functie (sectie 3) en de andere voor het shell commando printf (sectie 1):

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

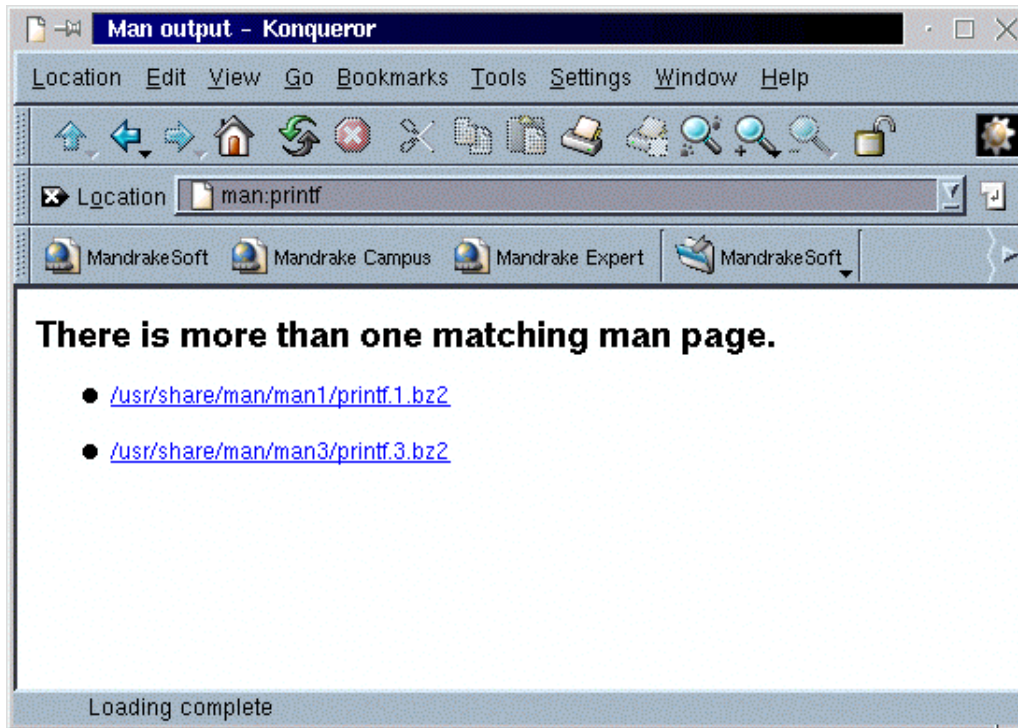
De verschillende secties zijn:

Sectie

- 1 User commands (gebruiker commando's)
- 2 System calls (functies die door de kernel geleverd worden)
- 3 Subroutines (functies geleverd door libraries (bibliotheken))
- 4 Devices (speciale bestanden in de /dev directory)
- 5 File format descriptions (indeling van bestanden, zoals bijvoorbeeld /etc/passwd)
- 6 Games (spellen, spreekt voor zich)
- 7 Miscellaneous (Diversen, bijvoorbeeld macro's, conventies)
- 8 System administration tools that only root can execute (tools voor het onderhoud van het systeem)
- 9 Another (overig)
- n New documentation (nieuwe documentatie, kan verplaatst worden naar een meer toepasselijke sectie)
- 1 Local (documentatie die refereert aan dit specifieke systeem)

Als je dus "man 1 printf" typt, krijg je de documentatie over het shell commando printf en "man 3 printf" levert de beschrijving van de C-library functie. Als je alleen "man printf" typt, wordt de eerstgevonden pagina getoond (meestal printf van sectie 1).

om te controleren of er meerdere versies van man pages zijn, kun je gebruik maken van het whichman commando, zoals hierboven weergegeven (te downloaden van mijn homepage of je typt "man:printf" in konqueror en het zal je het volgende vertellen:



MANPATH

Het man commando zoekt de man pages gebaseerd op de waarde van de omgevingsvariabele MANPATH. Helaas zetten veel Linux distributies dit pad verkeerd. Vaak is /usr/lib/perl5/man, welke een rijke set documentatie bevat over alle perl functies, niet opgenomen. Je kunt het opnemen in je MANPATH (in .bashrc of .tcshrc of ...) met:

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
```

Na het zetten van je man path kun je "man Pod::Man" proberen om te zien of je een van de pages over Perl krijgt.

De opmaak trefwoorden

Het schrijven van een man page is erg simpel. Het is een eenvoudige opmaak taal waarbij trefwoorden (keywords) voor de opmaak beginnen met een punt aan het begin van de regel. Deze trefwoorden worden ook wel macro's genoemd. De belangrijkste zijn:

```
.TH -> (Title/Header) Dit begint de titel/header van de man page  
.SH -> (Section Header) Sectie kop  
.PP -> Nieuwe paragraaf
```

." -> Een commentaar regel
.TP -> Spring de tekst in die 2 regels na deze macro komt

De syntax van .TH is:

.TH [naam van het programma] [sectie nummer] [centrale voettekst] [linker voettekst] [centrale koptekst]

De syntax van .SH is:

.SH tekst voor een kop

De syntax van .PP is ligt erg voor de hand. Het veroorzaakt een regeleinde.

Ik vind het soms handig om voor-geformateerde tekst op te nemen voor programma code voorbeelden. Dit kun je doen met:

```
.nf
_jouw_pre_formatted_
_tekst_komt_hier_____
.fi
```

Merk op dat dit groff/nroff macros zijn en daardoor niet bij de speciale man-page macros horen. Ze lijken echter goed te werken op alle Unix systemen.

Alle man-page opmaak macros zijn gedocumenteerd in de man-page genaamd groff_man(7) (Klik hier voor een html versie van de groff_man(7) page). Ik zal niet alle macros uitleggen, maar stel in plaats daarvan voor om de groff_man page te lezen. De groff_man page is erg gedetailleerd en bevat alles wat je moet weten.

De hoofdstukken

Voordat we beginnen met het schrijven van onze eigen page, zou je moeten weten dat man pages normaal gestructureerd worden in hoofdstukken. Bij conventie zijn de mogelijke titels als volgt:

NAME	Name sectie, de naam van de functie of commando
SYNOPSIS	Gebruik
DESCRIPTION	Algemene beschrijving
OPTIONS	Zou opties en parameters moeten bevatten
RETURN VALUES	Function calls, voor sectie 2 en 3 manpages
ENVIRONMENT	Beschrijft omgevingsvariabelen.
FILES	Bestanden geassocieerd met het onderwerp
EXAMPLES	Voorbeelden en suggesties
DIAGNOSTICS	Meestal gebruikt voor sectie 4 device interface diagnostics
ERRORS	Secties twee en drie fout- en signaal afhandeling
SEE ALSO	Kruisreferenties en citaten
STANDARDS	Conformaties aan standaarden indien van toepassing
BUGS	Problemen en beperkingen
SECURITY CONSIDERATIONS	

andere Beveiligings issues om rekening mee te houden
Eigen headers kunnen toegevoegd worden door de auteur

Een voorbeeld man-page

Hier is een korte voorbeeld man-page. Merk op dat de \- nodig is om de dash (-) te onderscheiden van de verbindingstreepjes. Typ dit alles in een tekst editor en sla het op als cdspeed.1.

Opmerking van de vertaler: onderstaande voorbeeld is vertaald, het is echter aan te raden man-pages in het Engels te schrijven, daar dit meer voorkomt.

```
.TH cdspeed 1 "10 September, 2003" "versie 0.3" "USER COMMANDS"
.SH NAME
cdspeed \- verlaag de snelheid van je cdrom om een snellere toegangstijd
te krijgen
.SH SYNOPSIS
.B cdspeed
[\-h] [\-d device] \-s snelheid
.SH DESCRIPTION
Moderne cdrom drives zijn te snel. Het kan meerdere seconden kosten
om een 60x cdrom drive op toeren te laten komen en data te lezen van
de drive. Het resultaat is dat deze drives een stuk trager zijn dan
een 8x of een 24x drive. Dit is vooral waar als je alleen af en toe
(bijvoorbeeld iedere 5 seconden) een klein bestand leest.
Deze utility limiteert de snelheid van de drive en laat de drive
sneller reageren bij het benaderen van kleine bestanden.
.PP
cdspeed maakt de drive ook minder luidruchtig en is erg bruikbaar als
je naar muziek wil luisteren op je computer.
.SH OPTIONS
.TP
\-h
geef een korte help tekst weer
.TP
\-d
gebruik het opgegeven device in plaats van /dev/cdrom
.TP
\-s
stel de snelheid in. Het argument is een geheel getal. Nul zet de maximum
snelheid terug.
.SH EXAMPLES
.TP
Zet de maximale snelheid naar die van een 8 speed cdrom:
.B cdspeed
\-s 8
.PP
.TP
Zet de maximum snelheid terug:
.B cdspeed
\-s 0
.PP
.SH EXIT STATUS
cdspeed levert een nul als exit status als het succesvol de maximum
snelheid van de cdrom drive heeft gezet. In geval van fouten wordt er geen nul
geleverd.
.SH AUTHOR
Guido Socher (guido (at) linuxfocus.org)
```

```
.SH ZIE OOK  
eject(1)
```

Klik hier ([cdspeed.html](#)) om bovenstaande pagina te bekijken (Engels).

Weergave en opmaak van je man-page

Tijdens het schrijven van je man-page, zou je deze van tijd tot tijd moeten bekijken om te controleren of hij er goed uitziet. Typ:

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

of

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

Om een man-page te converteren naar platte, voor-geformateerde tekst (bijvoorbeeld voor een spellingscontrole) gebruik:

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

Om te converteren naar Postscript (voor printen of verdere conversie naar pdf) gebruik:

```
groff -man -Tps jouw_manpagebestand.1 > jour_manpagebestand.ps
```

Om de man page te converteren naar html, gebruik:

```
man2html jouw_manpagebestand.1
```

Perl POD gebruiken om man pages te genereren

Ik weet dat veel mensen het vreemd vinden om een man page te bewerken in een tekst editor. Ze willen de man page genereren. Het Perl POD documentatie formaat is een goede keus. Je kunt de page in POD syntax schrijven en dan het volgende commando gebruiken:

```
pod2man jouw_manpagebestand.pod > jouw_manpagebestand.1
```

De syntax van de perl pod documentatietaal is beschreven in een man page genaamd perlpod. Het bovenstaande man page voorbeeld zou er in pod formaat uitzien zoals hieronder. Merk op dat POD gevoelig is voor spaties en de lege regels rond "=head" zijn ook nodig.

```
=head1 NAME
```

```
cdspeed - verlaag de snelheid van je cdrom om een snellere toegangstijd  
te krijgen
```

```
=head1 SYNOPSIS
```

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

=head1 DESCRIPTION

Moderne cdrom drives zijn te snel. Het kan meerdere seconden kosten om een 60x cdrom drive op toeren te laten komen en data te lezen van de drive. Het resultaat is dat deze drives een stuk trager zijn dan een 8x of een 24x drive. Dit is vooral waar als je alleen af en toe (bijvoorbeeld iedere 5 seconden) een klein bestand leest. Deze utility limiteert de snelheid en laat de drive sneller reageren bij het benaderen van kleine bestanden.

cdspeed maakt de drive ook minder luidruchtig en is erg bruikbaar als je naar muziek wil luisteren op je computer.

=head1 OPTIONS

B<-h> geef een korte help tekst

B<-d> gebruik het opgegeven device in plaats van /dev/cdrom

B<-s> stel de snelheid in. Het argument is een geheel getal. Nul zet de maximum snelheid terug.

=head1 EXAMPLES

Zet de maximum snelheid naar die van een 8 speed cdrom:

```
cdspeed -s 8
```

Zet de maximum speed terug:

```
cdspeed -s 0
```

=head1 EXIT STATUS

cdspeed levert een nul als exit status als het succesvol de maximum snelheid van de cdrom drive heeft gezet. In geval van fouten wordt er geen nul geleverd.

=head1 AUTHOR

Guido Socher

=head1 SEE ALSO

eject(1)

Referenties

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

<p>Site onderhouden door het LinuxFocus editors team © Guido Socher "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Vertaling info: en --> -- : Guido Socher (homepage) en --> nl: Guus Snijders <ghs@linuxfocus.org></p>
---	--