SelfLinux-0.13.1

Linux Hilfe

Autor: Dennis Roch (*ysae@users.sourceforge.net*)
Formatierung: Frank Börner (*frank@frank-boerner.de*)
Lizenz: GFDL

Inhaltsverzeichnis

1 Einleitung und Überblick

2 Die Option "--help"

3 man

3.1 Aufruf
3.2 Aufbau
3.2.1 Name
3.2.2 Syntax (Synopsis)
3.2.3 Beschreibung (Description)
3.2.4 Optionen (Options)
3.2.5 Dateien (Files)
3.2.6 Siehe auch (See also)
3.2.7 Fehler (Bugs)
3.2.8 Autoren (Authors)

4 info

5 Womit kann ich ... ?

5.1 apropos & Co
5.2 HOWTOs

6 Paketdokumentation

1 Einleitung und Überblick

Linux und die auf ihm laufenden Programmen sind zumeist extrem flexibel und mächtig, aber auch dem entsprechend umfangreich. Man vergisst deshalb schnell einmal, welche Bedeutung die eine oder andere Option hatte. Manchmal sucht man auch nur die passende Anwendung für seine Zwecke oder einem ist lediglich der genaue Name eines Programms entfallen.

Wenn man sich zu helfen weiß, ist alles halb so schlimm. Die gängigsten Wege, in solchen Fällen weiter zu kommen, sollen im Folgenden vorgestellt werden.

2 Die Option "--help"

Der Befehl `"Programmname" --help` liefert eine kurze,überblickartige Beschreibung des Programms. Dabei wird in der Regel auf den Zweck des Programms, Aufruf und die wichtigsten Optionen bzw. Tastenkombinationen eingegangen.

Tip: Die Ausgabe umfasst in der Regel eine Bildschirmseite. Falls der Text länger sein sollte, übergibt man die Ausgabe einfach dem Programm `less`. Der Befehl lautet dann `"Programmname" --help | less`. Nun sollte es möglich sein mit den Cursortasten zu scrollen. Man beendet `less` durch Druck auf die Taste `q`.

3 man

Nachdem `--help` eine eher spärliche Ausbeute brachte, ist es nun an der Zeit `man` (engl.: manual Handbuch) kennen zu lernen. Dieses Handbuch bietet detaillierte Informationen über die auf dem System installierten Programme.

Dabei bringt jedes (Programm-)Paket seine eigenen Handbuchseiten mit. Bei der Installation wird es zum bereits bestehenden Teil hinzugefügt.

3.1 Aufruf

Die Handbuch-Seite (manpage) über ein bestimmtes Programm kann man sich mittels

```
user@linux ~/ $ man N "Programmname"
```

anzeigen lassen.

`N` ist dabei eine Nummer, die angibt, in welchem Kapitel des Handbuches nachgeschlagen werden soll. In den meisten Fällen kann sie weg gelassen werden.

Von Bedeutung ist sie jedoch dann, wenn ein Programm in mehreren Teilen des Handbuchs beschrieben ist. Die Einteilung erfolgt standardmäßig in acht Kategorien, die der folgenden Liste oder

```
user@linux / $ man man
```

entnommen werden können.

1	Benutzerbefehle
2	Systemaufrufe
3	Funktionen aus Systembibliotheken
4	Spezielle Dateien (z.B. Geräte)
5	Dateiformate
6	Spiele
7	Makropakete und Konventionen (z.B. `man`)
8	Befehle für Systemadministration (i.d.R. nur für root)

Wird keine Zahl angegeben, wird die Manpage angezeigt, die zuerst gefunden wird.

Da wir es bei der Anzeige wieder mit `less` zu tun haben, kann man sich in den Seiten per Cursortasten bewegen und durch Druck auf die Taste `q` zur Shell zurückkehren.

Eine Suche lässt sich über `/"Suchtext"` starten. Um sich für den gleichen Begriff weitere Treffer anzeigen zu lassen, reicht ein Druck auf die Taste `n`. Die Fundstellen im Text sind jeweils hervorgehoben.

Wer mit diesen elementaren Funktionen nicht zufrieden ist, dem kann ein Hilfetext zu `less`, der über `h` zu erreichen ist, auf die Sprünge helfen.

Tip: Nicht jeder mag längere Texte am Monitor lesen. Besitzt man ein funktionierendes Drucksystem, so kann man die Manpage mit Hilfe des Kommandos

```
user@linux ~/ $ man -t "Programmname" | lpr
```

zu Papier bringen.

3.2 Aufbau

Die Gliederung einer typischen Manpage orientiert sich an gewissen Vorgaben.

Folgende Punkte findet man in den meisten Manpages (in Klammern sind die englischen Bezeichnungen angegeben):

3.2.1 Name

Wie der Name schon sagt, beinhaltet dieser Abschnitt den Namen des Programms. Außerdem wird noch eine einzeilige Kurzbeschreibung des Programms angegeben.

Für das Programm `apropos`, das wir in diesem Kapitel auch noch behandeln werden, würde das beispielsweise so aussehen:

"`apropos` - sucht die Manualkurzbeschreibung in der Indexdatenbank"

3.2.2 Syntax (Synopsis)

Sie beschreibt wie man das Programm aufruft, welche Angaben in jedem Fall gemacht werden müssen und welche zusätzlich gemacht werden können.

Für `apropos` sieht das wie folgt aus:

```
user@linux / $ apropos [-dhV] [-e|-r|-w] [-m System[,...]] [-M Pfad]
Schlüsselwort
```

Alles, was nicht in eckigen Klammern steht, muss angegeben werden. In unserem Beispiel ist dies zum einen der Programmname (logisch) und zum anderen ein Schlüsselwort (also Suchbegriff).

Die eckigen Klammern umschließen alle freiwilligen Optionen. Man kann also (muss aber nicht) `-d` oder `-h` oder `-V` bzw. beliebige Kombinationen dieser Buchstaben als Optionen verwenden (z.B. `-Vd`; das Minus muss nur einmal angegeben werden).

Alternative Optionen werden durch vertikale Striche getrennt. `[-e|-r|-w]` heißt also, man kann -eine- der drei Optionen angeben.

Die darauf folgenden Optionen verlangen beide die Angabe eines Arguments. Beispielsweise muss auf `-M` (wenn man es verwendet) eine Pfadangabe folgen.

3.2.3 Beschreibung (Description)

Hier wird der Sinn und Zweck des Programms erläutert und grundlegende Bedienprinzipien dargelegt.

3.2.4 Optionen (Options)

Bisher weiß man noch nicht unbedingt, was z.B. die Option `-e` bewirkt. Genau darüber wird man in diesem Abschnitt aufgeklärt.

3.2.5 Dateien (Files)

Hier sind die wichtigsten Dateien aufgelistet, die das Programm für seine Arbeit benutzt. Dazu findet man oft ein kurze Beschreibung, wofür die jeweilige Datei gut ist.

Es handelt sich zumeist um Konfigurationsdateien, aber auch mal um solche, in denen Datenbanken gespeichert sind oder ähnliches.

3.2.6 Siehe auch (See also)

In diesem Abschnitt wird auf Manpages zu anderen Programmen hingewiesen, die entweder mit dem jeweiligen Programm zusammen arbeiten oder aber ähnliche Funktionen erfüllen (und daher von Interesse sein könnten).

3.2.7 Fehler (Bugs)

Es werden bekannte Fehler des Programms, sowie Inkompatibilitäten und häufige Bedienungsfehler aufgelistet.

Manchmal findet sich auch ein Hinweis, an wen man sich wenden kann, wenn man glaubt einen noch unentdeckten Fehler gefunden zu haben.

3.2.8 Autoren (Authors)

Hier werden Lorbeeren an alle diejenigen verteilt, die wesentlich zur Entwicklung des Programms beigetragen haben.

Übrigens: Bisher war immer von Programmen die Rede. Darüber hinaus existieren aber auch noch Manpages für bestimmte Konfigurationsdateien. Z.B. liefert:

```
user@linux ~/ $ man fstab
```

Hinweise zum Aufbau von /etc/fstab.

4 info

Über das info-Hilfesystem findet man manchmal nur die selben Informationen wie mittels man. Nicht selten jedoch gehen die gebotenen Informationen weit über die der Manpages hinaus.

Vor allem bietet info ein anderes Bedienkonzept. Es wird die Möglichkeit der Verknüpfung von Texten (also des Setzens von Links) genutzt. Der Benutzer kann dabei durch eine Dokumenten-Hierarchie navigieren und sich Stück für Stück von generellen Aussagen ausgehend immer tiefer in das Thema einlesen.

Der Aufruf erfolgt über die einfache Befehlszeile

```
user@linux ~/ $ info "Programmname"
```

bzw.

```
user@linux ~/ $ info
```

Letztere Variante stellt einen Index der vorhandenen Themen zur Verfügung.

In jedem Falle werden in der oberen Zeile Informationen über die Hierarchie angezeigt. Von links nach rechts sind das: die zugehörige info-Datei, der Name des aktuellen Dokuments, der des nächsten (Link), der des vorhergehenden (Link) sowie ein Link eine Ebene aufwärts.

Eine Einführung in info liefert ein Druck auf die Taste h. Für die Navigation in den Dokumenten sind folgende Tasten von Bedeutung:

n	next	Springt zum der Hierarchie nach nächsten Dokument
p	previous	Tut das Selbe für das vorher gehende Dokument
u	up	Geht in der Hierarchie eine Ebene nach oben
Space und Backspace		Scrollen im aktuellen Dokument seitenweise herunter bzw. herauf
Cursortasten		Bewegung des Cursors bzw. normales Scrollen
Return/Enter		Wenn sich der Cursor über einer Verknüpfung (also einem Link) befindet, so wird zum zugehörigen Dokument gesprungen. In Texten werden Links meistens zwischen einen "*" und "::" eingeschlossen (also * Link_auf_irgendwas::).
s	search	Sucht nach dem Begriff, den man anschließend eingibt (mit abschließendem Return/Enter). Weitersuchen über erneute Suche ohne Angabe eines Suchbegriffs
q	quit	Beendet info

Interessant ist auch noch die Option --apropos. Mittels

```
user@linux ~/ $ info --apropos Suchbegriff
```

kann man die Info-Seiten nach einem bestimmten Thema durchsuchen - ähnlich dem zu man gehörenden Befehl apropos der im nächsten Abschnitt beschrieben wird.

5 Womit kann ich ... ?

5.1 apropos & Co

Manpages sind eine schöne Sache, aber was macht man, wenn man nur weiß, was man machen will, aber nicht, wie das entsprechende Programm heißt?

Hier schafft `apropos` Abhilfe. Es durchsucht die Manpages jeweils im Abschnitt Name, in dem (wie oben erwähnt) auch eine Kurzbeschreibung enthalten ist.

Wenn man mehrere Suchbegriffe übergibt so nimmt `apropos` eine ODER-Suche vor. Für

```
user@linux ~/ $ apropos linux help
```

würde `apropos` nach allen Programmnamen und -beschreibungen suchen die **linux** oder **help** oder beides beinhalten (also gar nicht so wenige).

Man kann natürlich auch nach dem dem Vorkommen beider Begriffe suchen (UND-Suche). Dazu sucht man mit `apropos` den ersten Begriff und filtert seine Ausgabe mit `grep` (nach dem zweiten Begriff).

Für alles das sowohl **linux** als auch **help** enthält, würden wir also

```
user@linux ~/ $ apropos linux | grep help
```

eingeben.

`grep` hat einige sehr interessante Optionen (`-v` sucht z.B. alles was den gegebenen Begriff nicht enthält) Für eine Befehlsübersicht sei auf `man grep` verwiesen.

Zum Schluss seien noch mal drei Optionen von `apropos` erläutert:

- `-e` sucht nach einer exakten Übereinstimmung mit dem Suchbegriff so findet man mit "string" nur noch "string" und nicht mehr "strings"
- `-r` der Suchbegriff wird als regulärer Ausdruck interpretiert
- `-w` Hier kann man mit so genannten Wildcards/Jokern rumspielen. Das wären zum Beispiel das `*` das für eine beliebige Zeichenkette oder das `?`, das für genau ein beliebiges Zeichen steht.
  Aber Achtung: gefunden wird nur das, was mit allen Wildcards auf einen ganzen Programmnamen oder eine ganze Beschreibung passt. (jeweils ein `*` am Anfang und am Ende des Suchbegriffs macht sich also ganz gut)

5.2 HOWTOs

Man & Co sind immer dann gut, wenn man Informationen zu einem bestimmten Programm sucht. Manche Aufgaben sind allerdings etwas komplexer und bedürfen somit einer intensiveren Erläuterung. Dafür sind die als Kurzreferenz gedachten Dokumentationen nicht ausgelegt.

Im Unix/Linux-Bereich haben sich daher umfangreichere Anleitungen unter dem Namen **HOWTO** etabliert.

Zu finden sind diese in der Regel im Verzeichnis /usr/share/doc/HOWTO oder auch /usr/doc/HOWTO (abhängig von der verwendeten Distribution).

Dort findet man die installierten HOWTOs in Verzeichnissen nach Sprache und Format sortiert. Für die Sortierung werden die Länderkürzel und die typische Dateierweiterung des Formats verwendet. So finden sich in de-html beispielsweise deutschsprachige Hypertextdokumente oder in en-txt englische Textdateien.

Die Dateien selber sind nach dem Thema benannt (manchmal findet sich auch die Sprache noch mal wieder). Beispielsweise DE-Drucker-HOWTO.html oder CD-Writing-HOWTO.

Sucht man neue und aktuelle HOWTOs, so lohnt sich ein Blick auf die Homepages des des "Linux Documentation Project" (englisch) http://www.tldp.org bzw. des "deutschen Linux HOWTO Projekts" http://www.linuxhaven.de/dlhp.

Tip: Vor allem die englischen Text-HOWTOs sind häufig nur als komprimierte Dateien installiert. Um sie zu lesen kann man folgendes Kommando verwenden:

```
user@linux ~/ $ zless dateiname.gz
```

6 Paketdokumentation

Als letzte Möglichkeit der Informationsbeschaffung bleibt noch der Blick in die Dokumentation der installierten Pakete.

Diese liegt meist unter /usr/share/doc. Je nach Distribution sind aber auch Pfade wie /usr/doc oder /usr/doc/packages möglich.

Was sich in den Verzeichnissen befindet, die nach den Paketen benannt sind, ist unterschiedlich. Allerdings tragen fast alle wichtigen Dateien aussagekräftige Namen und es findet sich auch fast immer eine Datei README. Sie gibt darüber Auskunft, welche Dateien oder Webseiten Informationen zum Programm/Paket bereitstellen oder tut dies selbst.