

2.2 Hilfe suchen über die Befehlszeile

Zertifikat:	Linux Essentials
Version:	1.6
Thema:	2 Sich auf einem Linux-System zurechtfinden
Lernziel:	2.2 Hilfe suchen über die Befehlszeile
Lektion:	1 von 1

Einführung

Die Kommandozeile ist ein sehr komplexes Werkzeug. Jeder Befehl hat seine eigenen Optionen, und darum ist **Dokumentation** der Schlüssel zu einem Linux-System. Neben dem **Verzeichnis `/usr/share/doc/`**, in dem die meiste Dokumentation liegt, bieten verschiedene andere Tools Informationen zur Verwendung von Linux-Befehlen. Diese Lektion konzentriert sich auf Methoden zum Zugriff auf diese Dokumentation, um Hilfe zu erhalten.

Es gibt eine Vielzahl von Methoden, um Hilfe auf der Linux-Befehlszeile zu erhalten: `man`, `help` und `info` sind nur einige davon. Für Linux Essentials konzentrieren wir uns auf `man` und `info`, da es sich um die am häufigsten verwendeten Hilfe-Tools handelt.

Ein weiteres Thema dieser Lektion ist das **Auffinden von Dateien**, wobei Sie hauptsächlich mit dem Befehl `locate` arbeiten werden.

Hilfe auf der Kommandozeile aufrufen

Eingebaute Hilfe

Mit dem **Parameter `--help`** aufgerufen, liefern die meisten Befehle eine kurze Übersicht zu ihrer Nutzung. Obwohl nicht alle Befehle diesen Schalter bereitstellen, ist es dennoch ein guter erster Versuch, mehr über die Parameter eines Befehls zu erfahren. Beachten Sie, dass die Anweisungen von `--help` im Vergleich zu anderen Dokumentationsquellen, die wir im weiteren Verlauf dieser Lektion besprechen werden, oft recht knapp gehalten sind.

Manpages

Die meisten Befehle bieten eine "Manual Page" oder kurz "Manpage". Diese Dokumentation wird in der Regel mit der Software installiert und kann mit dem Befehl `man` aufgerufen werden: Der Befehl, dessen Manpage angezeigt werden soll, wird `man` als Argument mitgegeben:

```
$ man mkdir
```

Dieser Befehl öffnet die Manpage für `mkdir`. Über die Pfeiltasten nach oben und unten oder die Leertaste navigieren Sie durch die Manpage — `q` schließt sie wieder.

Jede Manpage ist in maximal 11 Abschnitte unterteilt, wobei viele dieser Abschnitte optional sind:

Abschnitt	Beschreibung
NAME	Name des Befehls und kurze Beschreibung

Abschnitt	Beschreibung
SYNOPSIS	Beschreibung der Befehlssyntax
DESCRIPTION	Beschreibung der Wirkung des Befehls
OPTIONS	Verfügbare Optionen
ARGUMENTS	Verfügbare Argumente
FILES	Hilfsdateien
EXAMPLES	Ein Beispiel für den Einsatz des Befehls
SEE ALSO	Querverweise zu verwandten Themen
DIAGNOSTICS	Warn- und Fehlermeldungen
COPYRIGHT	Autor(en) des Befehls
BUGS	Bekannte Fehler und Beschränkungen des Befehls

In der Praxis enthalten die meisten Manpages nicht alle diese Teile.

Manpages sind in acht Kategorien organisiert, die von 1 bis 8 nummeriert sind:

Kategorie	Beschreibung
1	Benutzerbefehle
2	Systemaufrufe
3	Funktionen der C-Bibliothek
4	Treiber und Gerätedateien
5	Konfigurationsdateien und Dateiformate
6	Spiele
7	Verschiedenes
8	Systemadministrator-Befehle
9	Kernel-Funktionen (nicht Standard)

Jede Manpage gehört zu genau einem **Abschnitt**, mehrere Abschnitte können jedoch **Manpages mit gleichem Namen enthalten**. Nehmen wir beispielweise den Befehl `passwd`, mit dem man das Passwort eines Benutzers ändert: Da `passwd` ein Benutzerbefehl ist, befindet sich seine Manpage in **Abschnitt 1**. Neben dem Befehl `passwd` hat auch die Passwortdatenbankdatei `/etc/passwd` eine Manpage, die ebenfalls `passwd` heißt. Da es sich um eine Konfigurationsdatei handelt, gehört sie zu **Abschnitt 5**. Der Verweis auf eine Manpage enthält darum meist neben dem Befehlsnamen auch den Hinweis auf den entsprechenden Abschnitt, also beispielsweise `passwd(1)` oder `passwd(5)`.

Standardmäßig zeigt `man passwd` die erste verfügbare Manpage an, in diesem Fall `passwd(1)`. Die Kategorie der gewünschten Manpage ruft man etwa mit `man 1 passwd` oder `man 5 passwd` auf.

Wir haben bereits besprochen, wie man durch eine Manpage navigiert und wie man zur Kommandozeile zurückkehrt. `man` verwendet intern den Befehl `less`, um den Inhalt der Manpage anzuzeigen. `less` lässt Sie nach Text innerhalb einer Manpage suchen. Um nach dem Wort `linux` zu suchen, können Sie einfach `/linux` für die Vorwärtssuche ab dem Punkt, an dem Sie sich auf der Seite befinden, oder `?linux` verwenden, um eine Rückwärtssuche zu starten. Danach sind alle Treffer im Text markiert und die Seite springt zum ersten Treffer. In beiden Fällen springen

Sie mit `N` zum nächsten Treffer. Für Informationen zu weiteren Features drücken Sie `H`, und es erscheint ein umfangreiches Menü.

Info-Seiten

Ein weiteres Werkzeug, das Ihnen bei der Arbeit mit dem Linux-System helfen wird, sind die Info-Seiten, die in der Regel detaillierter sind als die Manpages und in Hypertext formatiert sind, ähnlich wie Webseiten im Internet.

Info-Seiten rufen Sie wie folgt auf:

```
$ info mkdir
```

Für jede Info-Seite liest `info` eine Info-Datei, die in einzelne Knoten innerhalb eines Baumes strukturiert ist. Jeder Knoten umfasst ein einfaches Thema, und der Befehl `info` enthält Hyperlinks, über die Sie von einem zum anderen gelangen.

Ähnlich wie `man` hat auch `info` Befehle zur Seitennavigation, über die Sie mehr erfahren, indem Sie auf einer Info-Seite `?` drücken. Diese Werkzeuge erleichtern Ihnen die Navigation und erklären, wie Sie auf die Knoten zugreifen und sich innerhalb des Knotenbaums bewegen.

Das Verzeichnis `/usr/share/doc/`

Wie bereits erwähnt, enthält das Verzeichnis `/usr/share/doc/` die umfangreichste Dokumentation der Befehle, die das System verwendet, sowie je ein Verzeichnis für die meisten der auf dem System installierten Pakete. Der Name eines solchen Verzeichnisses entspricht in der Regel dem Paketnamen, oft ergänzt um die Versionsnummer. Darin findet sich eine Datei `README` oder `readme.txt` mit der grundlegenden Dokumentation des Pakets. Neben der Datei `README` können weitere Dokumentationsdateien enthalten sein, wie etwa das Änderungsprotokoll (Changelog), das die Geschichte des Programms im Detail dokumentiert, oder Beispiele für Konfigurationsdateien.

Die Informationen in der Datei `README` sind von Paket zu Paket unterschiedlich. All diese Dateien sind im Klartext geschrieben, so dass sie mit jedem beliebigen Texteditor zu lesen sind. Anzahl und Art der Dateien hängen vom Paket ab. Schauen Sie sich einige dieser Verzeichnisse an, um einen Überblick über deren Inhalt zu gewinnen.

Dateien suchen

Der Befehl `locate`

Ein Linux-System besteht aus zahlreichen Verzeichnissen und Dateien und verfügt über viele Werkzeuge, um eine bestimmte Datei im System zu finden. Das schnellste ist der Befehl `locate`.

`locate` durchsucht eine Datenbank und gibt dann jeden Namen aus, der mit der angegebenen Zeichenkette übereinstimmt:

```
$ locate note
/lib/udev/keymaps/zepto-znote
/usr/bin/zipnote
/usr/share/doc/initramfs-tools/maintainer-notes.html
/usr/share/man/man1/zipnote.1.gz
```

Der Befehl `locate` unterstützt auch die Verwendung von Wildcards und regulären Ausdrücken, so dass die Suchzeichenfolge nicht mit dem vollständigen Namen der gewünschten Datei übereinstimmen muss. Mehr über reguläre Ausdrücke erfahren Sie in einer späteren Lektion.

Standardmäßig verhält sich `locate` so, als ob das Muster von Asteriks (*) umgeben wäre, so dass `locate MUSTER` das gleiche ist wie `locate *MUSTER*`. Das macht es möglich, Teilzeichenketten anstelle des genauen Dateinamens anzugeben. Sie können das Verhalten mit verschiedenen Optionen ändern, die in der Manpage zu `locate` erläutert sind.

Da `locate` aus einer Datenbank liest, finden Sie möglicherweise keine Datei, die Sie kürzlich erstellt haben. Die Datenbank wird von einem Programm namens `updatedb` aktualisiert. Normalerweise wird es regelmäßig ausgeführt, aber wenn Sie root-Rechte haben und die Datenbank sofort aktualisiert werden muss, können Sie den Befehl `updatedb` jederzeit selbst ausführen.

Der Befehl `find`

`find` ist ein weiterer beliebter Befehl zur Dateisuche. Er verfolgt einen anderen Ansatz als `locate`. `find` durchsucht einen Verzeichnisbaum rekursiv, einschließlich seiner Unterverzeichnisse. `find` führt eine solche Suche bei jedem Aufruf durch, verwaltet also keine Datenbank wie `locate`. Ähnlich wie `locate` unterstützt `find` auch Wildcards und reguläre Ausdrücke.

`find` benötigt mindestens den zu durchsuchenden Pfad. Darüber hinaus können so genannte Ausdrücke hinzugefügt werden, um Filterkriterien für die anzuzeigenden Dateien anzugeben, etwa der Ausdruck `-name`, der nach Dateien mit einem bestimmten Namen sucht:

```
~$ cd Downloads
~/Downloads
$ find . -name thesis.pdf
./thesis.pdf
~/Downloads
$ find ~ -name thesis.pdf
/home/carol/Downloads/thesis.pdf
```

Der erste `find`-Befehl sucht nach der Datei im aktuellen Verzeichnis `Downloads`, während der zweite nach der Datei im Heimatverzeichnis des Benutzers sucht.

`find` ist sehr komplex, daher wird er in der Prüfung Linux Essentials nicht behandelt; er ist aber ein mächtiges und nützliches Werkzeug in der Praxis.

Geführte Übungen

1. Nutzen Sie den Befehl `man`, um herauszufinden, was die folgenden Befehle bewirken:

Befehl	Beschreibung
<code>ls</code>	Zeigt den Inhalt eines Verzeichnisses.
<code>cat</code>	
<code>cut</code>	
<code>cd</code>	
<code>cp</code>	
<code>mv</code>	

Befehl	Beschreibung
mkdir	
touch	
wc	
passwd	
rm	
rmdir	
more	
less	
whereis	
head	
tail	
sort	
tr	
chmod	
grep	

- Öffnen Sie die Info-Seite von `ls` und finden Sie das MENU.
 - Welche Optionen haben Sie?
 - Finden Sie die Option, mit der Sie die Ausgabe nach dem Zeitpunkt der letzten Änderung sortieren.
- Zeigen Sie den Pfad zu den ersten 3 `README`-Dateien. Verwenden Sie den Befehl `man`, um die richtige Option für `locate` zu ermitteln.
- Erstellen Sie eine Datei namens `test` in Ihrem Home-Verzeichnis und finden Sie deren absoluten Pfad mit dem Befehl `locate`.
- Haben Sie sie sofort gefunden? Was mussten Sie tun, damit `locate` sie findet?
- Suchen Sie nach der Testdatei, die Sie zuvor erstellt haben, mit dem Befehl `find`. Welche Syntax haben Sie verwendet und wie lautet der absolute Pfad?

Offene Übungen

- Es gibt einen Befehl in der obigen Tabelle, der keine Manpage hat. Welcher ist es und warum, glauben Sie, gibt es keine Manpage?
- Erstellen Sie mit Hilfe der Befehle aus der obigen Tabelle den folgenden Dateibaum. Namen, die mit einem Großbuchstaben beginnen, bezeichnen Verzeichnisse, jene mit Kleinbuchstaben Dateien.

```

3. User
4. | Documents
5. | | Hello
6. | | | hey2
7. | | | helloa
8. | | | ola5
9. | | World
10. | | | earth9
11. | | Downloads
12. | | | Music
13. | | | Songs
14. | | | collection1
15. | | | collection2
16. | | Test

```

```
17. |      |
    | test | passa
```

18. Zeigen Sie auf dem Bildschirm das aktuelle Arbeitsverzeichnis an, einschließlich der Unterverzeichnisse.
19. Suchen Sie im Baum nach allen Dateien, die mit einer Ziffer enden.
20. Entfernen Sie den gesamten Verzeichnisbaum mit einem einzigen Befehl.

Zusammenfassung

In dieser Lektion haben Sie gelernt:

- Wie Sie Hilfe erhalten
- Wie Sie den Befehl `man` verwenden
- Wie Sie in der Manpage navigieren
- Verschiedene Abschnitte der Manpage
- Wie Sie den Befehl `info` verwenden
- Wie Sie zwischen verschiedenen Knoten navigieren
- Wie Sie nach Dateien im System suchen

Befehle, die in den Übungen verwendet werden:

man

Zeigt die Manpage an.

info

Zeigt die Info-Seite an.

locate

Durchsucht die `locate`-Datenbank nach Dateien mit dem angegebenen Namen.

```
find
```

Durchsucht das Dateisystem nach Namen, die einer Reihe von Auswahlkriterien entsprechen.

updatedb

Aktualisiert die `locate`-Datenbank.