

Von Befehlen zum Skript

Vi und Nano

Häufig genutzte Texteditoren

```
#!/bin/bash  
  
# This is our first comment. It is also good practice to document all scripts.  
  
echo "Hello World!"
```

- Linux-Anwendung arbeiten in einer Umgebung, in der selten grafische Texteditoren zur Verfügung stehen
- Bearbeitung von Textdateien über die Befehlszeile
 - ◆ vi
 - ◆ nano

Vi

```
#!/bin/bash  
  
# This is our first comment. It is also good practice to document all scripts.  
  
echo "Hello World!"
```

- Standardmäßig auf fast jedem Linux-System
- Klon → vi IMproved (vim)
- 3 verschiedene Modi
 - ◆ Navigationsmodus
 - ◆ Einfügemodus
 - ◆ Befehlsmodus

Vi

Navigationsmodus

- H - Cursor nach links
- J - Cursor nach unten
- K - Cursor nach oben
- L - Cursor nach rechts

Vi

Navigationsmodus

- W - springt zum Anfang des nächsten Wortes
- E - springt zum Ende des nächsten Wortes
- B - springt zum Anfang des vorherigen Wortes

Vi

Navigationsmodus

- 0 - Springt zum Anfang der aktuellen Zeile
- ^ - springt zum ersten nicht-leeren Zeichen der aktuellen Zeile
- \$ - springt zum Ende der aktuellen Zeile

Vi

Navigationsmodus

- Ctrl + u - Scrollt eine halbe Seite nach oben
- Ctrl + d - Scrollt eine halbe Seite nach unten
- Ctrl + b - Scrollt eine Seite nach oben
- Ctrl + f - Scrollt eine Seite nach unten

Vi

Navigationsmodus

- gg - springt zum Anfang der Datei
- G - springt zum Ende der Datei
- Ng - Springt zu Zeile n (z.B. 5g zu Zeile 5)

Vi

Navigationsmodus

- / - sucht nach einem Text nach vorne. Nach dem Eingeben des Suchbegriffs kannst du mit n weiter vorwärts und N rückwärts durch die Suchergebnisse springen
- ? - sucht nach einem Text rückwärts

Vi

```
#!/bin/bash  
  
# This is our first comment. It is also good practice to document all scripts.  
  
echo "Hello World!"
```

- Standardmäßig auf fast jedem Linux-System
- Klon → vi IMproved (vim)
- 3 verschiedene Modi
 - ◆ Navigationsmodus
 - ◆ Einfügemodus
 - ◆ Befehlsmodus

Vi

Wechsle in den Einfügemodus

- i - Wechselt in den Einfügemodus an der aktuellen Cursorposition (vor dem aktuellen Zeichen)
- I - Wechselt in den Einfügemodus am Anfang der aktuellen Zeile
- o - fügt neue Zeile unter der aktuellen ein und wechselt in Einfügemodus (O über)
- c: Löscht den Text, abhängig von dem folgenden Bewegungsbefehl (z.B. cw löscht bis zum Ende des aktuellen Wortes) und wechselt in den Einfügemodus.
- s: Löscht das Zeichen unter dem Cursor und wechselt in den Einfügemodus.
- S: Löscht die gesamte Zeile und wechselt in den Einfügemodus (entspricht cc).

Vi

Arbeit im Einfügemodus

- Sobald man im Einfügemodus ist, kann man Text eingeben, wie man es in einem Texteditor tun würde
- Backspace: Löscht das Zeichen links vom Cursor.
- Enter: Fügt eine neue Zeile ein.
- Ctrl + h: Funktioniert wie Backspace.
- Ctrl + w: Löscht das Wort links vom Cursor.
- Ctrl + u: Löscht den Text bis zum Zeilenanfang.
- Ctrl + j: Fügt eine neue Zeile ein (ähnlich wie Enter).
- Ctrl + o: Ermöglicht die Ausführung eines Befehls im Normalmodus, kehrt aber nach der Ausführung in den Einfügemodus zurück.
- Esc: Beendet den Einfügemodus und kehrt in den Normalmodus zurück.

Vi

Zurück in den Navigationsmodus

- Esc: Dies ist die häufigste Methode, um den Einfügemodus zu verlassen und in den Normalmodus zurückzukehren, wo du weitere Befehle ausführen kannst.

Vi

Zurück in den Navigationsmodus

Der Einfügemodus in Vim funktioniert ähnlich wie der Textbearbeitungsmodus in anderen Texteditoren. Der Hauptunterschied besteht darin, dass du in Vim explizit in diesen Modus wechseln musst, um Text einfügen oder bearbeiten zu können.

→ Esc: Dies ist die häufigste Methode, um den Einfügemodus zu verlassen und in den Normalmodus zurückzukehren, wo du weitere Befehle ausführen kannst.

Vi

Wechsel in den Befehlsmodus

- : (Doppelpunkt): Dies ist die häufigste Methode, um in den Befehlsmodus zu wechseln. Nachdem du : gedrückt hast, erscheint unten im Bildschirm eine Eingabezeile, in die du deinen Befehl eingeben kannst.

Vi

Befehlsmodus

- `:w` Speichert die Datei (write).
- `[Dateiname]`: Speichert die Datei unter dem angegebenen Dateinamen.
- `:q` Beendet Vim (quit).
- `:q!` Beendet Vim ohne zu speichern (force quit).
- `:wq` Speichert die Datei und beendet Vim.
- `oder ZZ`: Speichert die Datei (falls Änderungen vorhanden) und beendet Vim.
- `[Dateiname]`: Öffnet eine Datei zur Bearbeitung.
- `[Dateiname]`: Speichert die Datei unter einem neuen Namen und bleibt in der aktuellen Datei.

Vi

Aus den Befehlsmodus

- Nach der Eingabe eines Befehls drückst du die Enter-Taste, um den Befehl auszuführen.
- Wenn du während der Befehlsausführung den Befehl abbrechen möchtest, drücke Esc.

Nano

- neuer
- Einfacher zu bedienen als vi (?)
- Hat keine unterschiedlichen Modi
- Benutzer kann beim Start mit der Eingabe beginnen und verwendet Strg, um auf die am unteren Bildschirmrand gedruckten Werkzeuge zuzugreifen

Nano

Texteditoren sind eine Frage der persönlichen Präferenz

- neuer
- Einfacher zu bedienen als vi (?)
- hat keine unterschiedlichen Modi
- Benutzer kann beim Start mit der Eingabe beginnen und verwendet Strg, um auf die am unteren Bildschirmrand bedruckten Werkzeuge zuzugreifen