

Shell-Commands und Shell-Scripting

Internal reference: topics/01-2.md

Command Line

- Das CL-Tool (Terminal) ermöglicht es die Befehle der Bash abzusetzen. Dabei gibt es einige → [Short-Cuts](#).
- Der Standard-Prompt (Eingabeaufforderung) bildet sich nach dem Schema und kann geändert werden (\$PS1)
- Das Tilde-Zeichen ~ ist ein Kürzel für das Heimatverzeichnis (Home) des aktuellen Benutzers
- Die Heimatverzeichnisse befinden sich i.d.R. unter /home/<benutzername>.

Einzig Ausnahme: Das Heimatverzeichnis des Administratorusers (root) befindet sich unter /root/

- Das Semikolon ; trennt einzelne Befehle in einer Zeile voneinander (Aufreihung der Befehle: Sequenz)
- Das Pipe-Zeichen (AltGr-7) | verkettet einzelne Befehle in der Zeile (Piping)
- Ein Hashtag # leitet einen Kommentar ein.
- Mit einem \ am Ende einer Zeile lässt sich eine Befehlszeile auf der nächsten Zeile weiter fahren.
- Das \ ist auch das sog. ESC-Zeichen, welches dem nachfolgenden Zeichen seine spezielle Bedeutung enthebt oder eine gibt: \ ; , \t oder \\.
- Mit einem & am Ende eines Befehls führt den Befehl im Hintergrund aus. Konsole, bzw. Script fährt weiter ohne zu warten.

Help zu Kommandos

Nur in einem Native-Linux-System sind folgende Hilfskommandos verfügbar:

- Der Befehl man öffnet sie Hilfeseiten (manual) eines Befehls Syntax: man <Befehlsname>
- Der Befehl which findet den Ort eines installierten Programmes Syntax: which <Befehl>
- Der Befehl apropos durchsucht alle Hilfeseiten nach einem Stichwort Syntax: apropos <Stichwort> (nicht immer verfügbar)

Shell-Scripting Basics

Über die Bash Shell (Bourne Again Shell) als Shell Interpreter können nicht nur einzelne Kommandos, sondern ganze Shell-Scripts zur Automatisierung von Abläufen ausgeführt werden. Um das Skript auszuführen, müssen die Ausführungsberechtigung (mit `chmod +x filename`) gesetzt werden.

Die erste Zeile eines Shellskripts beginnt mit einem sha-bang #!, welcher nicht als Kommentar eingelesen wird, gefolgt von dem Pfad zum Shell Interpreter. Dieser Pfad sage dem Betriebssystem, dass diese Datei eine Liste von Befehlen ist, die dem angegebenen Interpreter gefüttert werden

sollen.

Bei falscher Pfad-Angabe nach dem sha-bang erscheint eine Fehlermeldung (zum Beispiel Befehl nicht gefunden). Es ist gang und gäbe ein Shellskript mit .sh enden zu lassen. Übrigens: Den korrekten Pfad kann mit `which bash` überprüft werden.

Die erste Zeile könnte so lauten:

```
#!/bin/bash**  
#  
# ...
```



Daniel Garavaldi

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m122_aws/topics/01_2?rev=1761855242

Last update: **2025/10/30 21:14**

