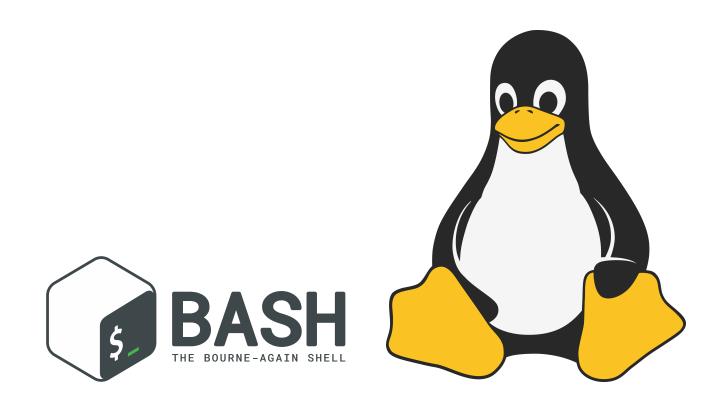
Arbeitsweise der bash



Inhaltsverzeichnis

- <u>Arbeitsweise der (interaktiven)</u> bash
- Verarbeitung der Kommandozeile
- Shell-Debugger

Arbeitsweise der (interaktiven) bash

- Lesen der (physischen) Kommandozeile
- Verarbeitung der Kommandozeile
- Ausführen der Kommandos
- Ausgabe des Prompts
- Warten auf die n\u00e4chste Eingabe

Diese Schritte wiederholen sich, bis die Shell (durch die Eingabe des Kommandos exit oder von Strg+D) beendet wird.

Der Vorgang ist allerdings etwas komplexer.

© 2025 Hermann Hueck 1/8

Verarbeitung der Kommandozeile

- Lesen der physischen Kommandozeile von der Standardeingabe (Dies ist normalerweise die Tastatur.)
- Zerlegung der physischen Kommandozeile in (evtl. mehrere)
 logische Kommandos. Dazu müssen die Kommando-Trenner (z.B. ;
) erkannt werden.
- Zerlegung der logischen Kommandos in "Wörter" (Tokens). Dazu müssen die Wort-Trenner (Leerzeichen) und die Maskierungszeichen (Anführungszeichen und Backslash) erkannt werden.

© 2025 Hermann Hueck 2/8

- Das erste Wort eines Kommandos ist das Kommandowort (der Name des auszuführenden Programms).
- Die übrigen Wörter sind die Argumente des Kommandos. Diese werden weiter verarbeitet:
 - Variablen-Substitution
 - Kommando-Substitution
 - Interpretation von Wildcards (wird später behandelt)
 - o etc.
- Durch den vorigen Schritt können sich die Argumente eines Kommandos verändern. Die Argumentliste wird neu aufgebaut.

© 2025 Hermann Hueck 3/8

- Ausführung des Kommandos:
 - Das Kommandowort ist ein Pfad: (Es enthält ein / -Zeichen):
 Ausführung der Programmdatei mit dem Pfad. Die Argumente werden übergeben.
 - Das Kommandowort ist kein Pfad: Das Kommando kann ein internes Kommando der Shell sein oder ein externes Programm, das als Datei im Dateisystem liegt.
 - (Aliase und Funktionen blenden wir zur Vereinfachung aus.)
 - Internes Kommando (Shell built-in): Die Shell führt das interne Kommando aus und übergibt die Argumente.

© 2025 Hermann Hueck 4/8

- Kein internes Kommando: Die Shell durchsucht von links nach rechts die Verzeichnisse, die in der Umgebungsvariablen PATH aufgeführt sind nach einer ausführbaren Datei mit dem Namen des Kommandos. Beim ersten Treffer wird das Programm ausgeführt und die Argumente werden übergeben.
- Wurde zum Kommandowort kein passendes internes oder externes Kommando gefunden, gibt die Shell eine Fehlermeldung aus.
- Wurde ein passendes Kommando gefunden, wartet die Shell auf das Ende der Ausführung.
- Ausgabe eines neuen Prompts (nur bei interaktiver Shell)
- Warten auf die nächste Eingabe von der Standardeingabe. © 2025 Hermann Hueck

5/8

Shell-Debugger

- Die Shell bietet einen eingebauten Debugger, der die fertig verarbeitete Kommandozeile vor der Kommandoausführung anzeigt.
- Den Debug-Ausgaben der Shell wird ein + -Zeichen vorangestellt.
- set -x aktiviert den Debugger.
- set +x deaktiviert den Debugger.

© 2025 Hermann Hueck 6/8

```
hermann@debian:~$ set -x # enable debugging
hermann@debian:~$ echo Hallo Welt
+ echo Hallo Welt
Hallo Welt
hermann@debian:~$ echo "Hier bin ich: $HOME"
+ echo 'Hier bin ich: /home/hermann'
Hier bin ich: /home/hermann
hermann@debian:~$ set +x # disable debugging
+ set +x
```

© 2025 Hermann Hueck 7/8

```
hermann@debian:~$ set -x # enable debugging
hermann@debian:~$ echo A comment: # this is the comment.
+ echo A comment:
A comment:
hermann@debian:~$ echo "No comment: # this is NO comment."
+ echo 'No comment: # this is NO comment.'
No comment: # this is NO comment.
hermann@debian:~$ set +x # disable debugging
+ set +x
```

© 2025 Hermann Hueck 8/8