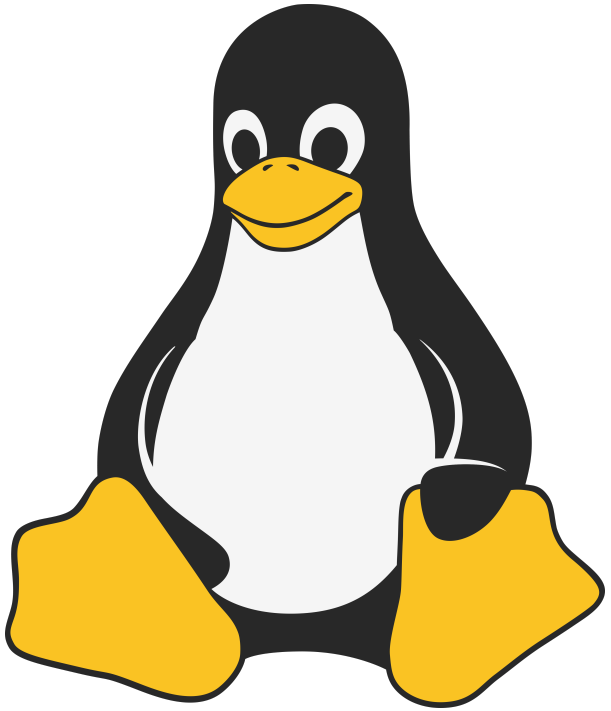
Shell-Praxis

Arbeitsweise der bash



Inhaltsverzeichnis

Arbeitsweise der (interaktiven) `bash`

- Lesen der (physischen) Kommandozeile
- Verarbeitung der Kommandozeile
- Ausführen der Kommandos
- Ausgabe des Prompts
- Warten auf die nächste 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.

Verarbeitung der Kommandozeile

- Lesen der physischen Kommandozeile von STDIN
- Zerlegung der physischen Kommandozeile in (evtl. mehrere) logische Kommandos. Dazu müssen die Kommando-Trenner (`;` und weitere) erkannt werden.
- Zerlegung der logischen Kommandos in "Wörter" (Tokens). Dazu müssen die Wort-Trenner (Leerzeichen) und die Maskierungszeichen (Anführungszeichen und Backslash) erkannt werden.

- 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 (wird später behandelt)
  - Interpretation von Wildcards (wird später behandelt)
  - etc.
- Durch den vorigen Schritt können sich die Argumente eines Kommandos verändern. Die Argumentliste wird neu aufgebaut.

- 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.
    - Internes Kommando (Shell built-in): Die Shell führt das interne Kommando aus und übergibt die Argumente.

- ○ ▪ 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 STDIN

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.

```
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
```

```
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
```