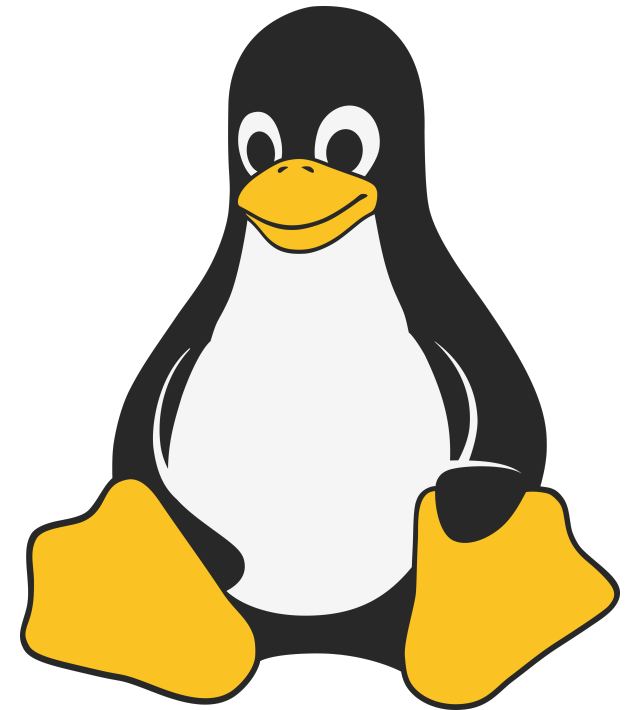


Arbeitsweise der **bash**



BASH
THE BOURNE-AGAIN SHELL



Inhaltsverzeichnis

- [Arbeitsweise der `bash` - vereinfacht](#)
- [Verarbeitung der Kommandozeile](#)
- [Shell-Debugger](#)

Arbeitsweise der `bash` - vereinfacht

- 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 von `Strg+D` oder des Kommandos `exit`) beendet wird.

Der Vorgang ist allerdings etwas komplexer.

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 (`;`, `|`, `&&`, `||`) 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:
 - Alias-Substitution für das Kommandowort
 - Variablen-Substitution
 - Kommando-Substitution
 - Interpretation von Wildcards (wird später behandelt)
 - etc.
- Durch den vorigen Schritt kann sich die Argumentliste 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.
 - (Aliase wurden schon ersetzt (s.o.). Funktionen blenden wir zur Vereinfachung aus.)
 - 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 der Standardeingabe.

Shell-Debugger

- Die Shell bietet einen eingebauten Debugger, der die fertig verarbeitete Kommandozeile vor der Kommandoausführung anzeigt.
- So lassen sich Alias-, Variablen- und Kommando-Substitutionen sowie die Interpretation von Wildcards sichtbar machen und überprüfen.
- 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
```

```
hermann@debian:~$ set -x # enable debugging
hermann@debian:~$ echo $(date)
+ echo 'Di 25. Mär 20:12:22 CET 2025'
Di 25. Mär 20:12:22 CET 2025
hermann@debian:~$ path
+ echo /usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/hermann/bin-trainer
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/hermann/bin-trainer
hermann@debian:~$ nl /etc/hostname
+ nl -ba /etc/hostname
    1  debian
hermann@debian:~$ set +x # disable debugging
+ set +x
```