HOWTO zur Unix-Philosophie

(C) 2006-2023 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC GmbH, http://www.ostc.de
http://www.ostc.de

$Id: unix-philosophy-HOWTO.txt,v 1.20 2025/02/23 20:14:55 tsbirn Exp $

Dieses Dokument beschreibt die Philosphie des Betriebssystems UNIX.

INHALTSVERZEICHNIS

1) Small is beautiful (KISS-Prinzip: Keep it simple, stupid)
2) Einfacher Datei- und Dateisystemaufbau
3) Mündiger Benutzer vorausgesetzt (weiß was er tut)
4) Kommandoorientierte Bedienung
5) Programmierbar/automatisierbar

---

1) Small is beautiful (KISS-Prinzip: Keep it simple, stupid)

* Kleiner KERN + viele getrennte kleine Dienstprogramme
* In Hochsprache "C" geschrieben (klein + schnell)
* Benutzerschnittstelle "SHELL" = ganz normales Programm
* "No news are good news" (keine Meldungen am Programmende)
* Programme lesen/schreiben von/auf Standard-Ein/Ausgabe (Pipeline)
* Werkzeugkasten-Prinzip (leichte Kombinierbarkeit auf Kommandozeile)
* Sysadmin "root" darf ALLES, normale Benutzer dürfen fast NICHTS
  (durchbrechbar mit Spezialrechten Set-UID/GID, Sticky und mit "sudo")
* Strenge Trennung von System- und Benutzerdaten
* Strenge Trennung von Betriebssystem, GUI und Anwendungen

2) Einfacher Datei- und Dateisystemaufbau

* Nur EIN hierarchischer Dateibaum mit Wurzel "/" (keine Laufwerke!)
* ALLES ist eine Datei (Speicher, Geräte,...; siehe Verz. /dev /proc /sys)
* JEDES Kommando/Programm kann mit Dateien arbeiten
* Datei = Folge von Bytes (keine Records fester Breite)
* Standard-Dateiformat = Zeilenorientierter ASCII-Text (Trennzeichen '\n')
* GROSS/kleinschreibung der Datei/Verz.namen wird unterschieden
* Alle Zeichen außer "\0" (NUL-Byte) und "/" (Verz.trenner) in Dateinamen erlaubt
* Rechtesystem einfach (3-stufige ACL: user/group/other + read/write/execute)

3) Mündiger Benutzer vorausgesetzt (weiß was er tut)

* Von Software-Entwicklern für Software-Entwickler
* Enthält standardmäßig Software-Entwicklungssystem (C, make, RCS)
* UNIX-Quellcode mitgeliefert (zumindest anfangs)
* Systemkonfiguration liegt dateibasiert unter "/etc" (analog Registry)
  + Remote-Administration per Editor und Systemkommandos möglich
  + Versionsverwaltung der Systemkonfiguration möglich
* Nichts wird verborgen (außer es ist unbedingt notwendig)
* Kooperatives Arbeiten wird vorausgesetzt
* Kaum Automatismen (z.B. Datenträger montieren)
* Kaum GUIs

4) Kommandoorientierte Bedienung

* Benutzerschnittstelle "SHELL"
* Saubere Trennung Datenkanal + Fehlerkanal (für Fehlermeldungen)
* GROSS/kleinschreibung der Kommandonamen + Parameter wird unterschieden
* Gesamte Dokumentation online verfügbar (Manual-Pages)
* Alles auf Englisch (Abkürzungen, Akronyme, Dokumentation, Fehlermeldungen)
* Kurze klein geschr. Namen für Kommandos/Dateien/Verz. (leicht zu tippen)

* Vermeidung von Sonderzeichen in Namen (z.B. Leerzeichen)
* Keine (wegzuklickenden) Erfolgsmeldungen, aber gute Fehlermeldungen

5) Programmierbar/automatisierbar
---
* Kommandoorientierte Bedienung
  + Einheitliche Kommandosyntax: CMD [-OPTIONS] SOURCE DESTINATION
  + Viele Kommandos
  + Kaum GUIs
* SHELL interaktiv + im Batch-Betrieb nutzbar
  + Vollwertige Programmiersprache
  + Einheitliche Quotierung: "...", '...' und \C
  + Einheitliche Namensexpansion: * ? [...] [^...] {...,...} ~
  + Einheitliche Kommando-Subst: `...` und $(...)
  + Einheitliche Umlenkbarkeit: Standard-Ein/Ausgabe < > 2> | >> 2>>
  + Einheitliche Exit-Stati: 0=Ok, >0=Fehler