

Unix

Eine kleine Einführung in der Unix Welt

Ein System von Entwickler für Entwickler

Entstehung von UNIX

- 1969 auf ein DEC PDP-7 (12 Bit Rechner)
- 1972–1974 Implementierung in C auf PDP-11/20 (16 Bit Rechner)
- 1977 BSD
- 1983-1990 Entwicklung von GNU (ohne Kernel)
- 1987 Minix 1.0 (Unixoid)
- 1991 Linux (Unixoid)
- 2000 Darwin, die Basis von MacOS
- 2003 ISO/IEC 9945:2003 Single UNIX Specification

POSIX

- ca. 1985 entstand POSIX mit den Ziel die verschiedenen entstandenen Implementierungen zu vereinheitlichen.

Paradigmen

- Alles ist eine Datei
- Vom Anfang an Multiuser und multitasking
- Halte alles einfach
- Kleine Programme, zusammengesetzt machen viel.
- Konfigurationsdaten werden in Textdateien festgehalten
- Script können schwierige Aufgaben erledigen

Paradigma alles ist eine Datei

- Die C-Funktionen **open()**, **write()** und **close()** können für (fast) alle Dateien und Geräte verwendet werden.
- Keine Laufwerke wie unter die andere Systeme der damaliger Zeit.

Die Shell

Die Schaltzentrale

- Vorläufer aller Shells ist die Thomson-Shell
- Die Bourne Shell wurde 1970 die Standard shell
- Die Posix-Shell wurde später normiert
 - Beispiel wäre beispielsweise die Korn-Shell
- Die Bourne-Again-Shell (bash) weicht an etliche Stelle von der Posix-Shell, kann aber in ein kompatibler Modus betrieben werden.
- MacOs verwendet die zsh (Z Shell)
- Debian basierte Systeme verwenden oft die dash (Debian Almquist Shell), sie ist Posix-konform und wäre schneller und robuster als die bash.

Scripts

- Sie werden von der verwendete Shell **`\${SHELL}`** aufgerufen
- Sie besitzen Anweisungen zur bedingte Ausführungen wie:

- if [Bedingung]; then Anweisung(en); elif Anweisung(en); fi
- while [Bedingung]; do Anweisung(en); done
- case \$Wort in; Wort1) Anweisungen;; Wort2 Anweisungen;; esac
- Die Ein-Ausgabe von Programme können mittels Pipes “|” weiterverarbeitet werden.
- Ein-Ausgabe Streams können auch mittels redirection von oder nach andere Quellen gelenkt werden.
- Funktionen können deklariert werden
 - function meineFunktion() { Anweisungen; }

Shebang

Die erste Zeile einen Shell Script beginnt oft mit ein Shebang:

```
#!/bin/sh
```

...

Mit: >

```
#!/usr/bin/python2
```

...

- Damit wird der Shell angewiesen das aufgeführte Kommando als Interpreter zu verwenden.
- Normalerweise sollte der Aufruf python python3 aufrufen.

System Konfiguration

- Sämtliche Dienste ein Unixartige System werden mit Dateien im Verzeichnis **/etc** konfiguriert
- Anwenderspezifische Konfigurationen unter
 - \$HOME/.config
 - \$HOME/.local
 - \$HOME/.<name>

Terminal basierten Editoren

- ed dürfte ein der ältesten Editor sein, es ist ein Zeileneditor
 - sed (Stream Editor) erlaubt es filter aufzubauen, die Syntax ist an ed angelegt
- vi, unter Linux vim wird nicht so oft benutzt ist aber sehr leistungsfähig und erlaubt meisten ein schnelleres Editieren als graphische Editoren
- GNU Emacs wird selten unter Linux verwendet.
- nano wird oft erwähnt, es ist für Anfänger leichter zu bedienen als vi.

Terminal unter graphische Oberfläche

- xterm war ein oder der erste Terminal Emulator, es wird kaum noch verwendet.
- Jede Desktop Umgebung liefer ein eigene Terminal
 - XFCE4: xfce4-terminal
 - Gnome: gnome-terminal
 - KDE: konsole
 - usw.

Lieferumfang einem Unix(oid) System

- Verwaltungs tool wie:
 - adduser,..., passwd, chmod chgrp usw.
 - mkfifo, mkdir, mknod
 - mount
 - chattr (Linux spezifisch)
 - fsck, mkfs, usw.
- Filter Programme wie:
 - sed, grep, awk, tr, ...
- alte Befehle wie:
 - test, cal, bc, dc, expr, read, more usw.

- diverse weitere Programme wie:
 - ls, cp, mv, ln, echo
 - diff comm,
 - tar, cpio, ar, zip, dd, ...

Vor- und Nachteil der Terminal basierten Tools

- Oft schneller zu bedienen
- Mehr Einstellmöglichkeiten
- Manchmal unübersichtlich

Graphische Oberfläche X

Netzwerk fähig, geeignet für “Thin Clients”

- 1984 Athena Projekt (MIT, DEC, IBM)
- 1986 X10.4
- 1987 X11.4
- **1988** Weiterentwicklung durch das X-Konsortium
- 1994 X11R6 (Open Source XFree86)
- 2003 X.org Foundation
- 2005 X11R7

Neue graphischen Grafische Oberfläche

- Seit ca. 2009 Entwicklung von Wayland

Toolkits

- Xaw
- Xt
- Motif / Lesstif
- xForms
- FLTK
- Qt
- GTK

Desktop

- CDE - Motif /X11
- KDE - QT / X11 + Wayland
- Gnome - GTK / X11 + Wayland
- Cinnamon - GTK / X11 + Wayland
- Mate - GTK / X11 + Wayland?
- XFCE - GTK / X11, für 4.20 voraussichtlich Wayland
- usw.

Demokratisierung

- Gnu (Gnu is Not UNIX) stellt weitgehend freie Alternative zu den verschiedenen UNIX Programme
- Der Code des Linux Kernel waren von Anfang an frei

Hard-Core Distributionen

- Slackware (Die älteste Distribution)
- Gentoo
- Archlinux

Weit verbreite Distributionen

- Viele Distributionen sind leicht zu installieren und beinhalten was das Herz begehrt.
 - Debian
 - Ubuntu
 - Mint
- Fedora
- Mageia
- Manjaro
- OpenSuse
- usw.

Distributionen für Rechenzentren und Cloud

- Debian
- Redhat
 - CentOS
- Suse (SLES)

Fragen?

Danke