Die Shell - Das Vorspiel

Die Shell, die Kommandozeile und ein Teil vom ganzen Rest

Dirk Geschke

Linux User Group Erding

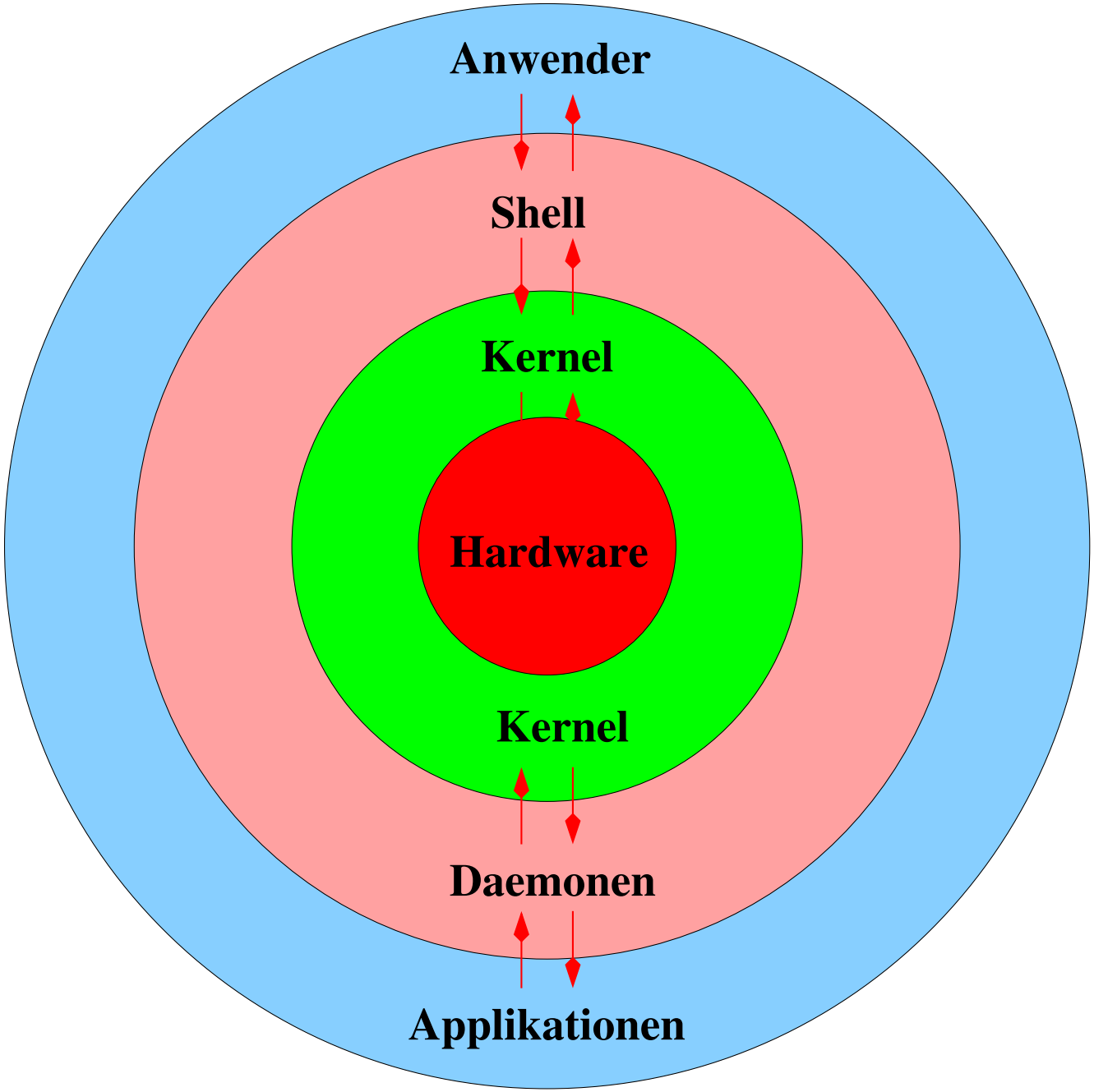
22. Oktober 2008

Gliederung

Grundprinzip: Kugel

1. innerste: Hardware
2. eingekapselt von: Kernel
3. umgeben von der Schale: Shell oder Daemonen
4. außen: Anwender oder Anwendungen
5. Kommunikation zwischen Kernel und Shell/Anwendungen via Systemcalls

Schematische Darstellung:



Allgemeines

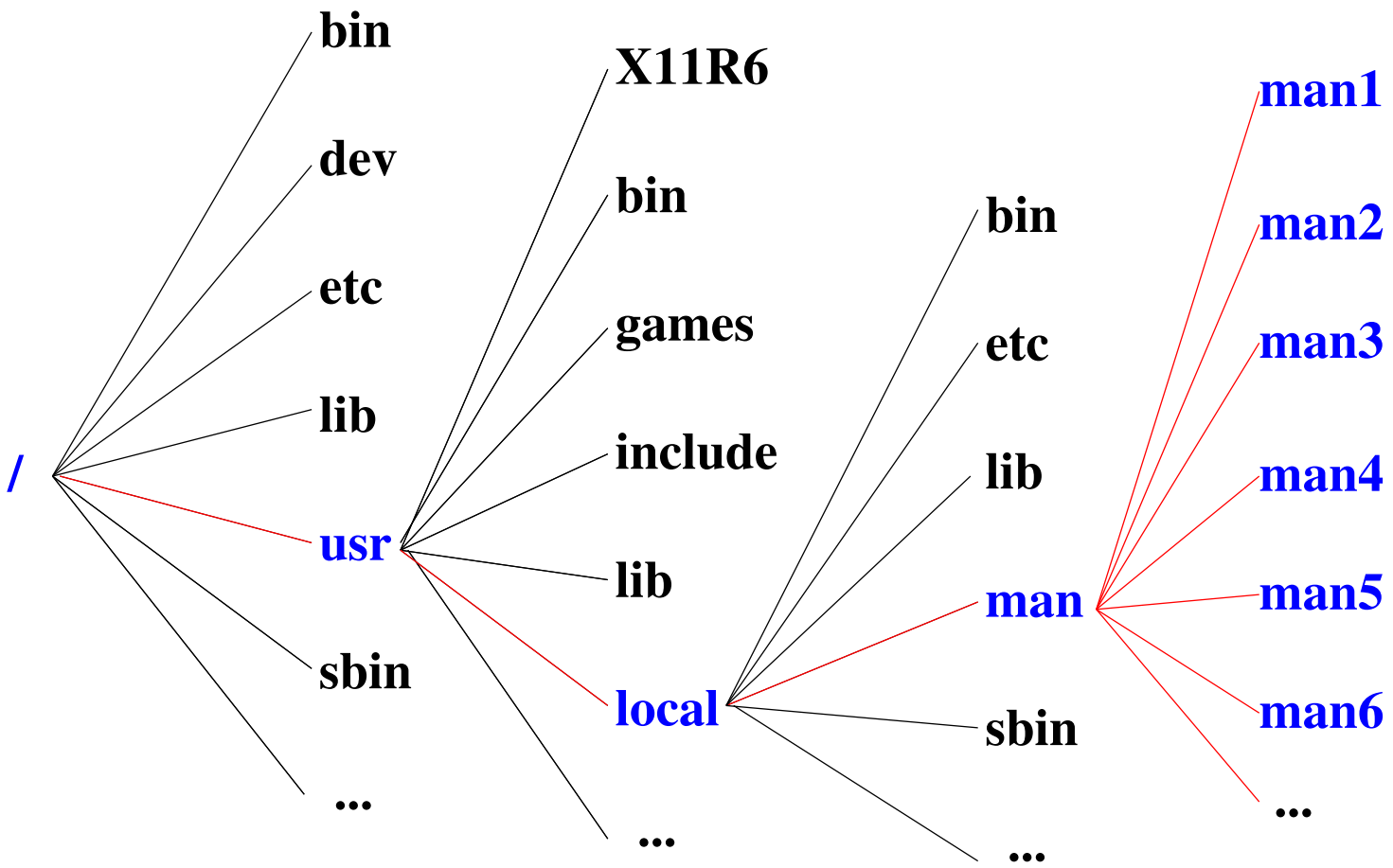
- Multitasking: paralleles Ausführen von Programmen
- Mehrbenutzersystem: mehrere Anwender können parallel arbeiten
- Dateiorientiert: *Alles ist eine Datei*
  - Hardware wird durch Gerätedateien devices dargestellt, z.B. die erste IDE-Festplatte heißt */dev/hda*, ein Drucker */dev/lp0*, erstes Pseudoterminal */dev/pts/1*
  - Ausnahme unter Linux: Netzwerkkarten
  - Festplatten oder Partitionen werden direkt auf Verzeichnisse montiert: ge–mount–ed.
  - Kommunikation via Filedeskriptoren: Unabhängig ob Datei oder Netzwerkverbindung

Unix-Philosophie

1. Write programs that do one thing and do it well.
   - bessere Optimierung der Programme möglich
   - weniger Nebenwirkungen
   - leichtere Fehlersuche
2. Write programs to work together.
   - Zusammenarbeit ermöglicht Modulsystem
3. Write programs to handle text streams, because that is a universal interface.
   - Texte lassen sich leichter lesen und analysieren

Verzeichnisbaum

4. Verzweigung, Basispfad: /usr/usr/local/man/



Verzeichnisse

/boot	Verzeichnis enthält idR Kernel und initrd
/bin	ausführbare Dateien für normale Benutzer
/dev	Gerätedateien
/etc	Konfigurationsdateien der Programme/Daemonen
/home	Benutzerverzeichnisse
/initrd	Initiale Ramdisk, kann beim booten für zusätzliche Module/Programme verwendet werden, liegt im Hauptspeicher
/lib	Kernel-Module, Bibliotheken die beim Booten benötigt werden $\longrightarrow$ `bin`, `sbin`
/mnt	temporärer Mountpoint zum Einhängen von Laufwerken
/opt	optionale Software

Verzeichnisse, Fortsetzung

/proc	Informationen über Prozesse
/root	Homeverzeichnis des Superusers
/sys	Kernelinformationen, `sysctl`
/tmp	temporäre Dateien
/usr	Programme für Benutzer
/usr/local	lokale Erweiterungen
/usr/share	Architekturunabhängige Dateien
/usr/share/man	Manual pages, Sektionen 1-9
/var	variables Verzeichnis $\longrightarrow$ veränderbare Dateien während des laufenden Betriebes

Arten von Dateisystemen

- Klassische Dateisysteme: minix, ext2, msdos
- Journaling Dateisysteme: ext3, jfs, reiserfs, xfs
- Memory Dateisysteme: tempfs, shfs
- Pseudo–Dateisysteme: procfs, devfs, usbfs, sysfs
- Identifikation von Dateien via Inodes
  - Lage der Datei
  - Eigentümer und Zugriffsrechte
  - Größe, Modifikationszeiten
  - Art der Datei
  - Verweiszähler

Arten von Dateien

Zeichen in erster Spalte der Ausgabe von `ls -l`:

- – reguläre Datei
- d Verzeichnis (directory)
- b blockorientiertes Device, z.B. Festplatte
- c zeichenorientierte (character) Device, z.B. Terminal
- l symbolischer Link, Verweis auf andere Datei
- p Pipe, FIFO: First In, First Out
- s socket: Netzwerkschnittstelle

Links

Es gibt zwei Arten von links:

hard link verweist auf den gleichen inode

- identische Dateien, müssen auf einer Partition liegen
- wird ein Link geändert so auch der andere
- wird ein Link gelöscht, so bleibt der andere erhalten
- prominente Beispiele: . und ..

symbolischer link zeigt symbolisch auf andere Datei

- Ziel muss nicht existieren
- kann über Partitionsgrenzen hinaus verwendet werden

Arten von Benutzer

root Superuser, darf (fast) alles

user normale Benutzer

- eigene Verzeichnisse, Dateien
- eigene Prozesse
- abgeschirmt von anderen Benutzern

daemon Hintergrundprozesse

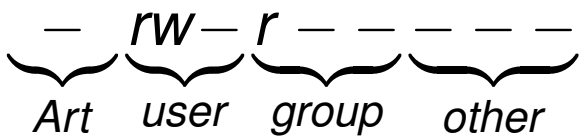
- keine Superuser-Rechte
- eingeschränkte Rechte, dürfen nur das notwendigste
- idR kein Einloggen möglich

Benutzer, Gruppen und der Rest

- Benutzer werden über `/etc/passwd` definiert
- Passwörter in `/etc/shadow`
- Gruppen werden in `/etc/group` definiert
- Zugriffsrechte für Dateien: Benutzer, Gruppe, Rest

  r lesbar
  w schreibbar
  x ausführbar

  Beispielsweise:



- Erweiterte ACLs sind auch möglich: `setfacl`

Ausgabe von `ls -l`

Beispiel:

```
-rw-r--- 1 root shadow 941 Oct 14 11:30 /etc/shadow
crw-rw-rw- 1 root root 1, 3 Oct 20 20:07 /dev/null
```

1. Dateiart: *-, b, c, d, l, p, s*
2. Zugriffsrechte: Eigentümer, Gruppe, andere
3. Verweiszähler: link counter
4. Eigentümer
5. Gruppe
6. Dateigröße, bei Devices: *major, minor*
7. Zeitpunkt der letzten Änderung
8. Dateiname

Sonderfälle

SUID	ausführender einer Datei erhält Rechte des Eigentümers
SGID	ausführender einer Datei erhält Rechte der Gruppe
Sticky Bit	Nur Dateieigentümer darf die Datei löschen
chmod	Setzen der Rechte, z.b. `chmod g+rw Datei`
chown	Setzen des Eigentümers
chgrp	Setzen der Gruppe