

Programmieren unter LINUX

Eine Einführung

46. Weinheimer UKW-Tagung 2001, Weinheim

Dipl.-Ing. Erich H. Franke, DK6II
Königsbach-Stein
email: dk6ii@afusoft.de

Ein nicht ganz zutreffender Vergleich

LINUX

- GNU general public license
- Quellcode offen
- Werkzeuge frei verfügbar
- Entwicklung:
 - Kernel: zentral
 - Zubehör: dezentral
- geringere Programmvielfalt am Markt
- weniger Office-Tools

aber

- Werkzeuge frei verfügbar

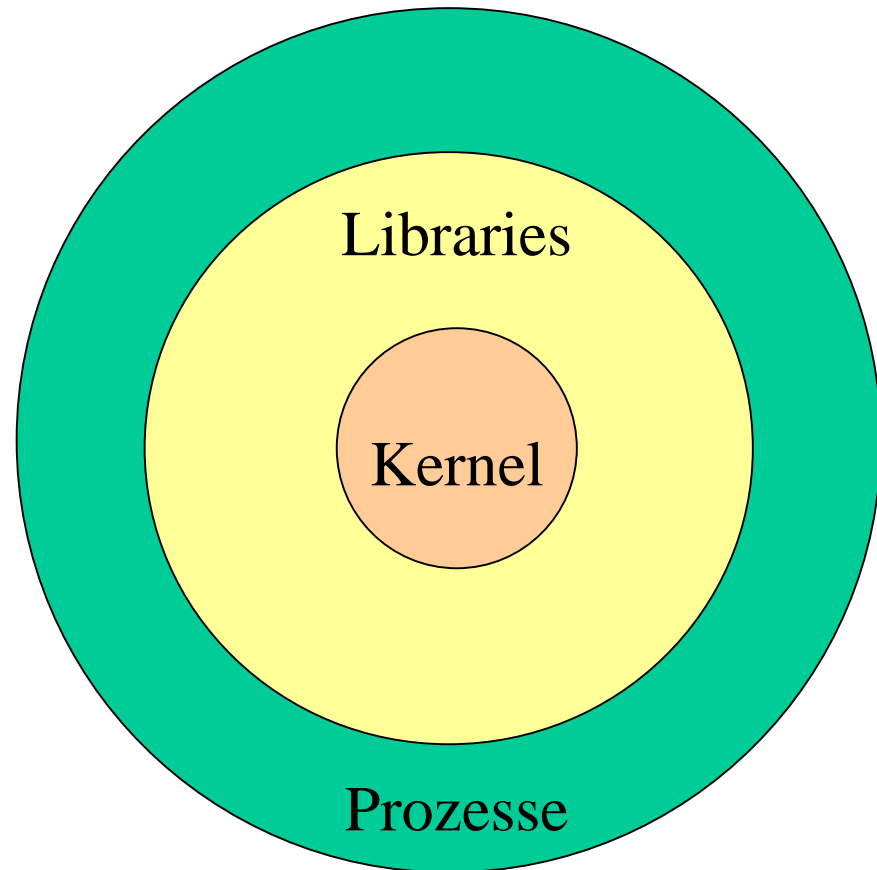
WINDOWS™

kommerzielles Copyright
nein
Werkzeuge kostenpflichtig
Entwicklung zentral
große Anzahl von Software am Markt verfügbar
Viele Office Tools



Prinzipieller Aufbau des Betriebssystems

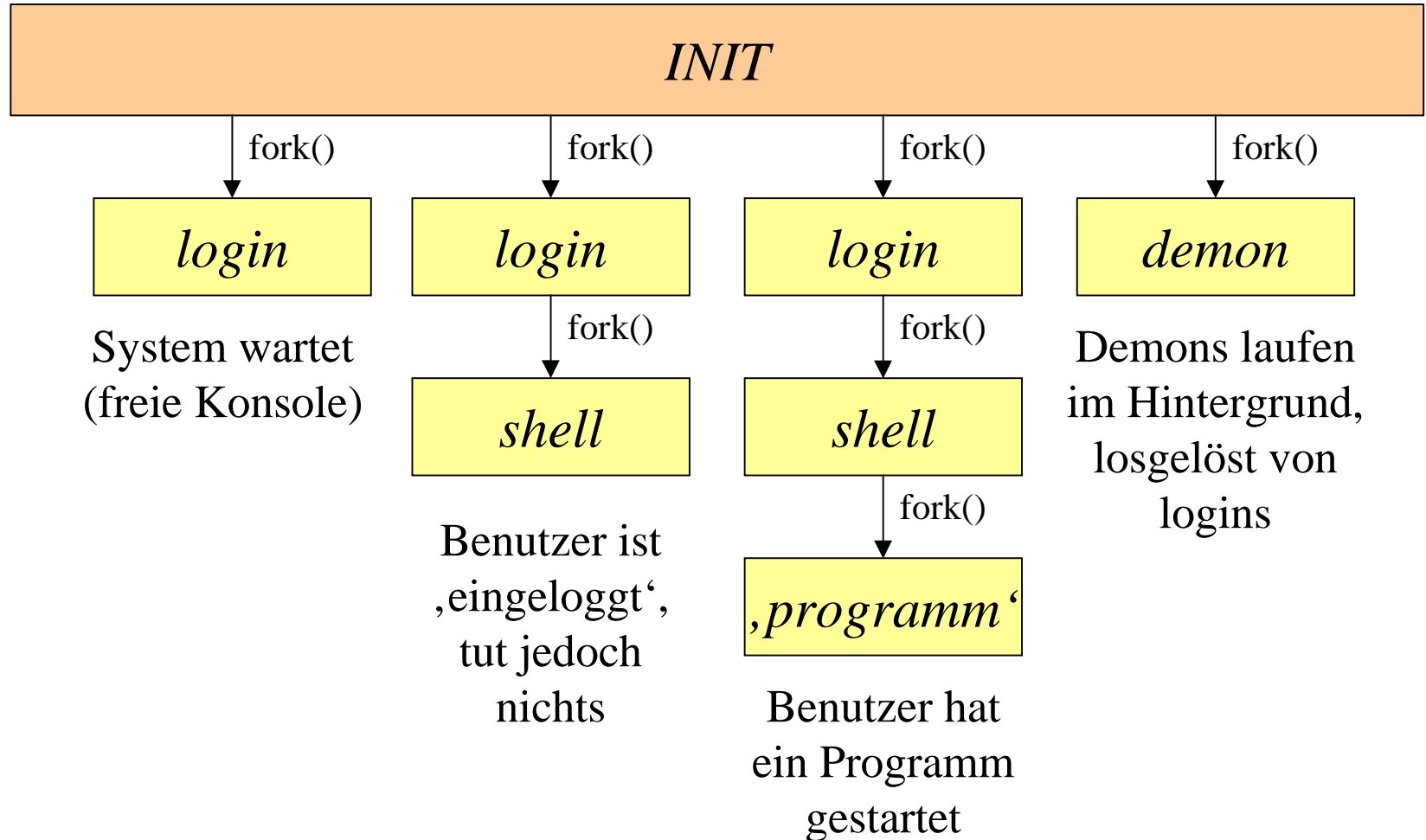
- Kernel
 - Scheduler
 - Memory manager
 - Device drivers
- Bibliotheken
 - Shared libraries
- Prozesse
 - Init
 - Systemprozesse
 - Shells
 - Anwenderprozesse



(Fast) alles ist ein Prozess

- Was ausgeführt wird, ist ein Prozess
- Prozesse können weitere (Kind-)Prozesse erzeugen (*fork*)
- Der oberste Prozess heißt *init*.
- Auch die ‚Kommandozeile‘ (*shell*) ist ein Prozess
- ‚Anwenderprogramme‘ sind einfach Kinder der Shell
- Auch die Benutzeranmeldung (*login*) ist ein einfacher Prozess
- Das, was Prozesse dürfen, ist über Rechte (*permissions*) des Benutzers geregelt unter dem sie laufen
- Prozesse können über klare Schnittstellen miteinander kommunizieren

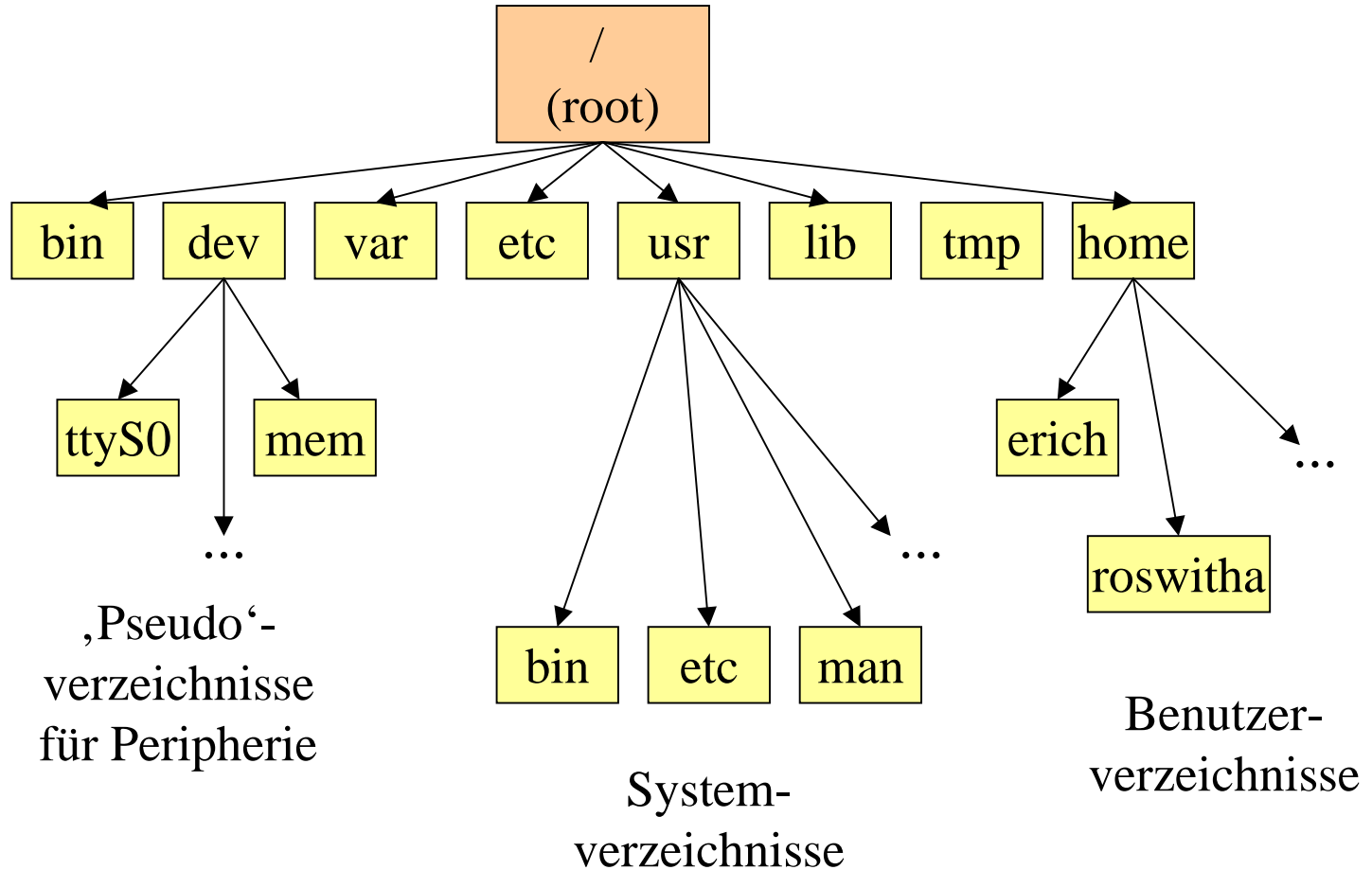
Prozesse bestimmen die Abläufe



(Fast) alles andere sind Dateien

- Massenspeicher sind als hierarchisches Dateisystem implementiert. (Es gibt verschiedene Arten, z.B. *reiserfs*)
- Das Dateisystem ist nicht ganz ungefährlich...
- ...hat aber eingebaute (hoch entwickelte) Schutzmechanismen
- Auch die Peripherie und Kommunikation sind als (Pseudo)-Dateien realisiert (Beispiel *dev/ttyS0*)
- selbst die Interprozess-Kommunikation ist so implementiert (z.B. *pipes*).
- Diese Objekte belegen keinen Festplattenplatz, können jedoch mit Standard I/O-Befehlen angesprochen werden

Daten und Dateien



Informationsquellen

- Arbeit mit LINUX erfordert Lernbereitschaft und gute Nerven
- ‚fast alle‘ Informationen können in den On-Line-Handbüchern (*man* pages) gefunden werden
- Nur: man findet sie nicht immer leicht, da die *man*-pages in verschiedene *sections* gegliedert sind. (z.B. *man read*)
- mit *apropos* findet man oftmals mehr als man möchte
- eine gute Quelle sind die *howto*‘s
- Es gibt inzwischen auch recht gute Literatur über

Programmierwerkzeuge

- Exzellentes C und C++ (GNU gcc und egcc)
 - sehr gute Codegenerierung, auch cross-platform
 - sehr gute Fehlerbehandlung
- perl zur Stringbearbeitung
 - mächtiger Sprachumfang
 - kryptischer Befehlssatz, steile Lernkurve, ‚organisch gewachsen‘
- Shell-Skripte zur Automatisierung
 - viel umfangreicher als bei MS-DOS (nicht vergleichbar)
- java und php
 - im web-bereich. Vor allem *php* für server-seitige Applikationen
- Viele weitere Sprachen und Assembler verfügbar

Graphik - Das kann ich auch... :)

- Grapische Oberflächen verfügbar (mehrere ‚leicht inkompatible‘ Interfaces)
- ‚Aufgesetztes‘ Design, nicht im Betriebssystem integriert
- Office-Tools für Hardliner ok, ansonsten...
- X Windows (XFree86) kämpft (noch) mit Performance und Security
- Video-Projekte (z.T. sehr gute) gibt es auch
- Aber: Sie brauchen Experimentierfreude und starke Nerven!

Cross the border



- LINUX exzellent als Plattform für Web-Server verwendbar (*Apache* ist Spitze!)
- Integration in WindowsTM-Netze mit *Samba* funktioniert sehr gut. Dadurch Kombination zwischen Windows-Oberflächen und LINUX-Apps möglich
- Integration in AppleTalk-Netzwerke mit *nettalk*
- Kompatibilität mit bestehenden UNIX-Plattformen besteht selbstverständlich
- Eher zum Spaß gibt es jede Menge Emulatoren: MS-DOS, C64 (!), sogar Windows

Communication (breakdown)

- Netzwerke mit tcp/ip sind Standard unter LINUX
- Sehr gute Implementierungen aller bekannten (und manch unbekannter) Dienste (z.B. ftp)
- Mehr und mehr secure com und firewalls
- Sehr einfache Programmierung im Netzwerkbereich
- Sogar AX.25 kernelseitig bereits implementiert

- Aber: Die Intrusion- und Security-Welle hat LINUX noch gar nicht richtig erreicht

Verletzbarkeit

- LINUX hat (UNIX-typische) Sicherheitsmechanismen auf Datei-, Task- und User-Level
- Viren sind aufgrund der Source-Verteilung eher selten
- Jedoch Gefahren durch ‚Trojaner‘ und ‚Würmer‘
- ‚Open Source‘ hat Vor- und Nachteile:
 - Angreifer können gezielt nach Lücken und ‚Exploits‘ suchen
 - Verteidiger können Programme im Source analysieren
- Sehr gute News-Groups (z.B. *securityfocus*)
- Aber: Wer spielt schon alle zwei Tage neue Patches ein?

Verzeichnisse und Dateitypen

- Eine normale Datei

```
-rw-r----- 1 erich users 27549 Sep 7 19:57 mtserv.c
```

- Ein Verweis (*link*) auf eine Datei

```
lrwxrwxrwx 1 erich users 12 Sep 7 20:03 parsusr.c -> ../parsusr.c
```

- Ein Unterverzeichnis (*directory*)

```
drwxr-xr-x 2 erich users 4096 Sep 5 21:57 Debug
```

- Eine Systeminformation (Pseudo-Datei)

```
-r--r--r-- 1 root root 0 Sep 7 20:15 /proc/uptime
```

- Ein Gerät für zeichenweisen Zugriff (*character device*)

```
crw-rw-rw- 1 root uucp 4, 64 Jan 19 2001 /dev/ttyS0
```

- Ein Gerät für Blockzugriffe (*block device*)

```
brw-rw---- 1 root disk 3, 1 Jan 19 2001 /dev/hda1
```