

# Linux Essentials

LPI 010 Training

Vortragender: Jürgen Wurzer



Mühlehner & Tavolato GmbH

# Topics

- ▶ Linux Community & Open Source
- ▶ Befehlszeile und Arbeiten mit dem Dateisystem
- ▶ Komprimieren, Suchen, Textbearbeitung
- ▶ Shellskripts
- ▶ Grundbegriffe Hardware, Kernel, Prozesse
- ▶ Netzwerkgrundlagen
- ▶ Benutzer- und Gruppenkonten sowie Berechtigungen

# Unix und Linux Geschichte und Versionen

\* 1970: **Unix** (00:00:00, Jan 1, 1970 is time zero for UNIX) [Ken Thompson, Dennis Ritchie, Brian Kernighan]

... different implementation paths: **System V**, **BSD**

\* 1982: **GNU Project** [Richard Stallman]



\* 1988: **Unix System V Release 4 (SVR4)** [AT&T and SUN Microsystems]



\* 1991-94: **Linux** (an open source Unix Kernel) [Linus Torvalds]



## Linux Distributions:

Linux Kernel + **GNU Tools** + Desktop-Tools + „PreConfiguration“

- Android und ChromeOS (Google)
- Red Hat (IBM), Fedora, CentOS(-Stream), OracleLinux
- SUSE (Novell), OpenSUSE
- Debian, Ubuntu, Kubuntu, Raspbian
- ....

## Unix Systems:

Solaris  
AT&T Unix  
SGI Unix  
HP Ux  
Ultrix,.....

## BSD

Apple:  
Mac OSx

# Linux, Linux-Distributionen und seine vielseitigen Einsatzbereiche

- ▶ Welche Linux-Distributionen gibt es?
  - Debian, Ubuntu, Fedora, openSUSE, RHEL (Red Hat Enterprise Linux), CentOS, Linux Mint, ...
- ▶ Was sind Linux Embedded Systeme?
  - minimalistische, auf spezifische Hardware-Funktion zugeschnittene Linux-Systeme die fest in einem Gerät verbaut sind.
- ▶ Wie werden Linux Embedded Systeme eingesetzt?
  - Alltagsgeräte: Router, Smart-TV, Fahrzeug-Infotainment, Smartphone (Android), Raspberry PI
  - Industriellen Steuerungen, Automobil-, Medizin- und Militäranwendungen

# Linux, Linux-Distributionen und seine vielseitigen Einsatzbereiche

- ▶ Verschiedene Einsatzmöglichkeiten von Android
  - Android = spezielles Linux-System
  - Smartphones, Tablets, Smartwatches, Fahrzeuge, TV-Boxen
- ▶ Verschiedene Einsatzmöglichkeiten des Raspberry Pi
  - kostengünstiger, kreditkartengroßer Einplatinencomputer
  - Zum Programmieren lernen, als Mediacenter, kleiner Server (Webserver, Sambaserver), Bastelprojekte in der Hausautomatisierung, Retro-Gaming-System (z.B. RetroPie)
- ▶ Was ist Cloud Computing?
  - Bereitstellung von IT-Ressourcen wie Server, Speicher, Datenbanken und Software über das Internet (die „Cloud“) auf Abruf
- ▶ Welche Rolle spielt Linux beim Cloud Computing?
  - Ist das dominierende Betriebssystem in der Cloud
  - Meistens die Grundlage für Server, Container und Virtualisierungstechnologien

# Linux Desktopanwendungen

- ▶ **Office** (LibreOffice oder Apache OpenOffice (beides Weiterentwicklungen von SUN StarOffice))
  - Writer, Calc, Impress, Draw, Base, Math  
→ Entspricht: Word, Excel, PowerPoint, Visio, Access, Formeleditor
- ▶ **Browser** (Mozilla–Firefox, Google–Chromium)
- ▶ **Email Client** (Thunderbird)
- ▶ **Multimedia–Anwendungen**
  - Blender (3D–Grafiksoftware), GIMP (Bildbearbeitung), Inkspace (Vector–Grafik–Editor), Audacity (Audio–Editor), ImageMagic (Bildbearbeitung)
- ▶ **Medienwiedergabe** (VLC, smplayer, Audacious,...)

# Linux Serverprogramme

- ▶ Webserver (Apache, Nginx, lighttpd)
- ▶ Datenbank (MySQL, MariaDB, PostgreSQL, ...)
- ▶ Datenfreigabe (NFS, Samba)
- ▶ Authentifizierungsserver (AD, Samba)
- ▶ Cloud-Lösungen (ownCloud, Nextcloud)
- ▶ Netzwerkdienste (DHCP, NTP, DNS, ...)
  
- ▶ Programmiersprachen (Java, C, C++, JavaScript, Python, PHP, C#, Bash, Powershell)

# Freie Software, OpenSource

- ▶ Freie Software (Richard Stallmann, 1985, GNU):  
*sehr ideologisch, politisch, sozial:*
  0. Ausführen wie man will und wozu man will
  1. Untersuchen und Anpassen wie man will
  2. Beliebige Weitergabe
  3. Beliebig Weiterentwickeln und wieder Freigabe  
(Pflicht!)
- ▶ Open Source  
*Offener Quellcode, ev. mit Einschränkungen zur Verwendung*  
→ verschiedene Lizenzen

# Lizenzen (FLOSS free/libre open soure sw)

- ▶ **GPL** (striktes Copyleft Prinzip)
  - Lizenz überträgt sich immer komplett auf Veränderungen
  - Anpassungen/Abschwächungen:
    - LGPL (Lesser GPL) erlaubt Mix mit unfreier SW
    - AGPL (Affero GPL) regelt Verkauf gehosteter SW
    - FDL (GNU Free Document License)
- ▶ **OSI** (OpenSource Initiative) permissive Prinzip
  - permissive = freizügig/erlaubend → max. Freiheit/Flexibilität
  - „OSI-Approved Lizenzen“
  - BSD (ganze Gruppe von Lizenzen, die fast nichts einschränken, OpenSource, den ich verändern, verwerten und dann auch als Closed Source vermarkten darf.
  - MIT-Lizenz, Apache-Lizenz, zlib-Lizenz, ...
- ▶ **CC** Creative Commons
  - Open Source Prinzip für andere Bereiche

# Desktop-Umgebungen

- ▶ Gnome (KISS, GTK, C)
- ▶ KDE (komplexer, Qt, C++)
- ▶ Terminalprogramme (Gnome Terminal, KDE Konsole, xterm, ...)
- ▶ Virtuelle TTYs: Strg+Alt+F# (F1, F2, etc.)
- ▶ Präsentationssoftware (Libre-Impress, LaTeX-Beamer, Reveal.js)
- ▶ Projektmanagement-SW (GantProject, ProjectLibre)

# Linux Server Einsatzgebiete

## ▶ Server

- Webserver, Datenbank, Dateiserver, Mailserver
- Netzwerkdienste (SSH, DNS, DHCP)

## ▶ Cloud

- IaaS – Infrastructure as a Service
  - Virtuelle Hardware
  - Virtualisierung (Xen, KVM, VirtualBox)
- PaaS – Platform as a Service
  - Fertige Umgebung / Platform
  - Heroku (Programmausführung ohne eigener VM)
- SaaS – Software as a Service
  - Fertige Software über den Browser oder eine App
  - Dropbox, Salesforce (meist Browserinterface)

# Datenschutz (beim Surfen)

- ▶ Cookie Tracking
- ▶ DNT (do not track) nur ein Wunsch!
- ▶ Privates Fenster
- ▶ Passwörter
  - Verwendung eines Passwort-Managers
  - KeePass (speichert verschlüsselt auf Platte)
  - BitWarden (speichert verschlüsselt in Cloud)
- ▶ Verschlüsselung
  - TLS/SSL
  - GnuPG
  - Festplattenverschlüsselung
    - Blockverschlüsselung (dm-crypt mit LUKS)
    - Stacked (EncFS, VeraCrypt [gibt's auch auf Windows und Mac])

Unix/Linux-Nutzer lieben die Kommandozeile ;-)

# Kommandozeile, Synonyme:

- ▶ command-line
  - ▶ CLI = command-line interface
  - ▶ Shell (Interpreter z.B: Bash, Zsh)
  - ▶ Terminal (ist das Fenster)

# Syntax Konvention für Kommandozeile:

## *Kommando Optionen Argumente*

*Is* *Optionen und Argumente sind optional*

*ls -l -s /etc /bin/\*a\* normalerweise Optionen zuerst*

*ls -- -la* -- beendet Optionenliste!! , -la' ist hier ein Argument

**ls --all**      **--longFormatOption** (benutzerfreundliche Version  
z.B. eine Alternative zu **-a**  
verfügbar mit den meisten *GNU tools*)

# Metazeichen & Quoting

- ▶ Metazeichen/Sonderzeichen haben eine besondere Bedeutung
  - | < > & ; # \* ? [ ] \$ ~ ( ) \ ' "

- ▶ Quoting
  - Aufhebung der Sonderbedeutung
  - Backslash \ hebt nächstes Sonderzeichen auf
    - Beispiel: \\*asd\\asd\'\\$
  - Zwischen Single Quotes ' hebt alles auf
    - Beispiel: 'asd"sad\$asd\*?'
  - Zwischen Double Quotes "
  - Beispiel: "asd\*?sd'as\$ {hugo} as\\$d`kom`sd"

Variable u Kommando wird NICHT aufgehoben!

# Shell Befehlstypen

## ▶ Shell Builtin

- cd, pwd, echo, exit

## ▶ Funktion

- Siehe Beispiel

## ▶ Alias

- Kurzname bzw. Abkürzung für einen Befehl
- Erzeugen mit: alias Kurzname='Befehl'
- Anzeigen: alias
- Löschen: unalias Kurzname
- Dauerhaft: ~/.bashrc oder ~/.bash\_aliases

## ▶ Hashed Kommando

- Die shell merkt sich verwendete externe Kommandos in einer Hash-Tabelle
- hash ... Kommando zum ausgeben der Tabelle

## ▶ Externes Kommando

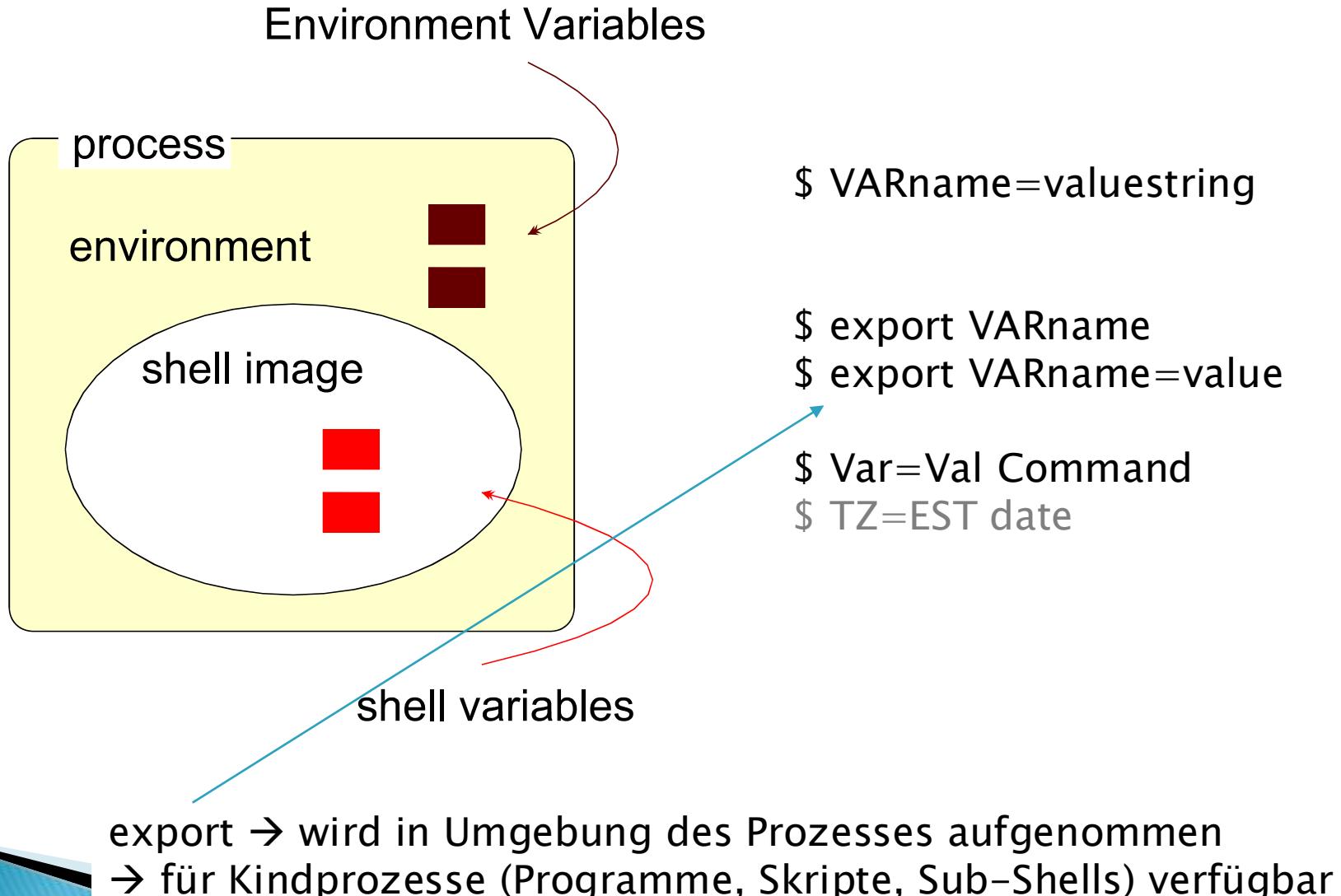
- ls, cp, ln, ...
- Die meisten Kommandos sind extern

```
meine_funktion() {  
    echo "Ich bin eine Funktion"  
    ls -l  
}  
  
# Aufruf der Funktion  
meine_funktion
```

*type Kommando*

Gibt den Befehlstyp aus (extern oder buildin)

# Variables and Environment



# Variablen der BASH

- ▶ **PATH**
  - Suchpfad für Befehle
- ▶ **HOME**
  - Standard-Benutzerverzeichnis
- ▶ **MAIL**
  - Speicherort der E-Mails (Posteingang)
- ▶ **MAILCHECK**
  - Prüfintervall für neue E-Mails (in Sekunden)
- ▶ **PS1**
  - Primärer Shell-Prompt (z. B. \$ oder #)
- ▶ **PS2**
  - Sekundärer Shell-Prompt (bei mehrzeiligen Befehlen)
- ▶ **TERM**
  - Terminal-Typ oder -Emulation (z. B. xterm)
- ▶ **TZ**
  - Zeitzoneneinstellungen des Systems

# Variablen der BASH

- ▶ \$\$
  - Enthält die Prozess-ID (PID) der aktuellen Shell-Instanz
- ▶ \$-
  - aktuelle Optionen der Shell. z. B. h für History-Erweiterung, i für interaktive Shell
- ▶ \$!
  - Speichert die Prozess-ID (PID) des zuletzt im Hintergrund (&) gestarteten Befehls.
- ▶ \$?
  - Enthält Exit-Status (Rückgabewert) des letzten Befehls
    - Wert 0 = Erfolg.
    - Wert ungleich Null (oft 1 bis 255) = Fehler

# Hilfe unter Linux

- ▶ **man**
  - Zeigt die Handbuchseiten (Manpages) für Befehle an
- ▶ **apropos (catman)**
  - Sucht in den Beschreibungen aller Manpages nach einem bestimmten Stichwort. Dies ist sehr nützlich, wenn man den genauen Befehl nicht kennt.
- ▶ **info**
  - Zeigt die Dokumentation im Info-Format an, die oft detaillierter und besser verlinkt ist als Manpages.
- ▶ **/usr/share/doc**
  - Dies ist das Standardverzeichnis im Linux-Dateisystem, in dem zusätzliche Dokumentation (z. B. README-Dateien, Beispiele) für installierte Software-Pakete gespeichert wird.

# Dateisuche unter Linux

- ▶ **locate *pattern***
  - Findet Dateien schnell anhand eines Musters, indem es eine zuvor erstellte Datenbank durchsucht, anstatt das Dateisystem live zu durchsuchen
- ▶ **updatedb**
  - Aktualisiert Dateinamen-Datenbank. Normalerweise automatisch ausgeführt. DB wird von locate verwendet.
- ▶ **find *verzeichnis***
  - Durchsucht das Dateisystem rekursiv nach Dateien
- ▶ **find -name *pattern***
  - Durchsuchen mit Suchkriterium

# Arbeiten mit Verzeichnissen

- ▶ Dateitypen
- ▶ Versteckte Dateien/Verzeichnisse
- ▶ \$HOME, ~
- ▶ absolute und relative Pfade
  
- ▶ Die wichtigsten Befehle + Optionen!
  - tree, ls, cd, pwd, mkdir, rmdir

# Arbeiten mit Dateien

- ▶ Dateinamens-Regeln
- ▶ einfaches Globbing
  - \* 0 oder mehr beliebige Zeichen
  - ? Genau ein beliebiges Zeichen
  - [1-4a-f]
  - [^1-4a-f]
  - [[:digit:]], [[:alpha:]], .....
- ▶ Befehle
  - cp, mv, rm, ln, touch, find

# Dateien Archivieren und Komprimieren

- ▶ tar
- ▶ gzip, gunzip, bzip2, bunzip2, xz, unxz
  - + die tar Optionen dazu -z, -j, -J
  - + die kombinierten Tools: *zgrep*, *xzgrep*....
- ▶ zip, unzip (Windows ZIP Format)

# Umleitung und Pipes

- ▶ 1> File
- ▶ 2> File
- ▶ &> File
- ▶ 1>> File
- ▶ 2>> File
- ▶ &>> File
- ▶ 0< File
- ▶ << hugo
  - ....
  - hugo
- ▶ <<< String
- ▶ Command | Command
- ▶ ls -l | head -20 | wc -w
- ▶ Filterprogramme:
  - cut, cat, find, less, head, tail
  - sort, wc, grep

# Reguläre Ausdrücke

simple regular expressions:

a  
\a  
. {  
[charset]  
[^charset]  
^ \$  
\(regexp\)...\\1

grep

extended regular  
expression addons:

+  
?  
(..)  
|

# Einfache Skripts

- ▶ Ausführungsvarianten und nötige Rechte
- ▶ #! Shebang
- ▶ Argumente \$\*, \$# , \$@, \$1, \$2, ...
- ▶ if – und Grundprinzip des Test-Kommandos
- ▶ exit
- ▶ for

# Betriebssystemauswahl

- ▶ Vergleich mit Windows, Unix, Apple
- ▶ Linux Distributionen:
  - Enterprise Linux-Distributionen
    - Red Hat Enterprise Linux
    - ~~CentOS~~, Oracle Linux, Rocky Linux, AlmaLinux
    - SUSE Linux Enterprise Server
    - Debian GNU/Linux
    - Ubuntu LTS
  - Consumer Linux-Distributionen
    - Fedora
    - Ubuntu non-LTS
    - openSUSE
  - Experimentelle und Hacker-Linux-Distributionen
    - Arch
    - Gentoo

# Hardware Grundlagen und Befehle

- ▶ **lshw** – Listet die gesamte Hardware auf. sudo lshw → noch mehr Details
- ▶ **lscpu** – Informationen über die CPU
- ▶ **lsblk** – Zeigt Informationen zu allen Blockgeräten (Festplatten, Partitionen, USB-Sticks)
- ▶ **lspci** – Listet alle angeschlossenen PCI-Bus Geräte auf.
- ▶ **lsusb** – Details zu angeschlossenen USB-Geräten
- ▶ **free -h** – Zeigt die Speicherauslastung (freien und belegten RAM sowie Swap-Speicher)
- ▶ **df -h** – Zeigt den freien Speicherplatz aller eingehängten Laufwerke an ("disk free").
- ▶ **uname -a** – Zeigt grundlegende Systeminformationen an, einschließlich des Kernel-Namens und der Version.
- ▶ **dmidecode** – Liest Hardwareinformationen direkt aus der DMI-Tabelle (BIOS/UEFI) aus (erfordert sudo).

# Dateisystemaufbau

## Programmverzeichnisse:

\$ *which cp*

- ▶ /bin
- ▶ /sbin
- ▶ /usr/bin
- ▶ /usr/sbin
- ▶ /usr/locale/bin
- ▶ /usr/locale/sbin
- ▶ ~/bin
- ▶ ~/.local/bin

~/.\*

- ▶ .profile, .....

## /etc:

*„Systemconfig-DB“*

- ▶ passwd
- ▶ shadow
- ▶ group
- ▶ hostname
- ▶ hosts
- ▶ profile, bash.bashrc
- ▶ nanorc
- ▶ resolv.conf
- ▶ sysctl.conf
- ▶ logrotate.conf
- ▶ crontab
- ▶ apt/sources\*, ...

*oder Verzeichnisse dafür:*

*\*.d*

# Dateisystemaufbau

- ▶ Teile, die den Kernel ausmachen
  - /boot/\* und /lib/modules/Version/\*
- ▶ GRUB
- ▶ systemd oder init Konfigurationen
- ▶ /proc !!!
- ▶ /dev !!!
- ▶ Speicher und SWAP

# Prozesse

- ▶ top
- ▶ ps
- ▶ /proc/PID
- ▶ uptime
- ▶ syslog und die wichtigsten Dateien in /var/log
- ▶ dmesg (Kenrel-RingBuffer)
- ▶ systemd-journald: journalctl

# Netzwerk Grundkonfiguration abfragen

- Internet, Netzwerk, Router
- Abfragen der DNS-Client-Konfiguration
- Abfragen der Netzwerk-Konfiguration
  - ▶ • route, ip route show
  - ▶ • ifconfig, ip addr show
  - ▶ • netstat, ss
  - ▶ • /etc/resolv.conf, /etc/hosts
  - ▶ • IPv4, IPv6, Sockets
  - ▶ • ping, dig, ss
  - ▶ • host

# Sicherheit in Linux – Benutzerkonten

## Grundprinzip

- ▶ **UID/GUI** (Nummern sehr Distro-abhängig)
  - 0: root
  - <100 oder <500: System Accounts
  - >1000 Standarduser und Service Accounts
- ▶ Dateien (im Detail!)
  - passwd, shadow
  - group, gshadow
  - sudoers / sudoers.d
- ▶ Befehle:
  - passwd, id, who, w, sudo, chsh, chfn

# Benutzer verwalten

- ▶ /etc/passwd, /etc/shadow, /etc/group, /etc/skel/
- ▶ useradd, groupadd
- ▶ userdel, groupdel
- ▶ passwd
  
- ▶ adduser, addgroup
  - Verfügbar bei Debian-basierten Distributionen
  - Benutzerfreundliche Wrapper für useradd und groupadd

# Dateiberechtigungen

Alles ist eine Datei! → Benötigen nur Dateiberechtigungen!

Permission Dateirechte	File Datei	Directory Verzeichnis	Device / Spec. File Gerätedatei
r	read, copy	read (name + InodeNr)	read
w	write, change	write (create, rename, <b>delete</b> )	write
x	execute	use Inodes of dir-entries (what can you do without??)	---

# Standard Permissions

## Standard Zugriffsrechte

user	group	other
r w x	r w -	r - -
1 1 1	1 1 0	1 0 0
7	6	4



process-euid == file-uid



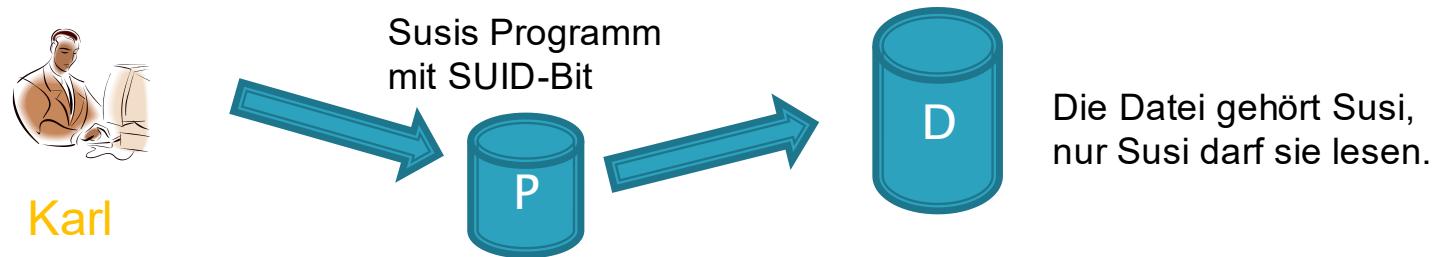
process-euid != file-uid &&  
process-egid == file-gid



euid != file-uid &&  
egid != file-gid

# SetUID-Bit: Rechte für Sonderaktionen durch Wechsel der effektiven Benutzer-ID

*Karl* ruft ein Programm auf, das *Susi* gehört und bei dem das Set-User-ID-Bit gesetzt ist  
→ wenn Karl nun eine Datei öffnet, die Susi gehört, wird Susi als zugreifende Identität verwendet.



Wenn der Besitzer eines „SUID-Programms“ root ist, erlaubt der Betriebssystemkern alles. Damit verlagert sich die gesamte Verantwortung für die Sicherheit vom Kernel-Code auf den Programmcode!

# Spezielle Permission Bits auf Verzeichnisse (FS abhängig)

## ▶ Sticky bit

- Mit Sticky Bit: Jeder kann zwar Dateien erstellen, aber nur der Besitzer der Datei (oder root) darf diese löschen oder umbenennen.

## ▶ SetGUID Bit

- Mit SetGUID-Bit: Jede neue erstellte Datei und Unterverzeichnis, in diesem Ordner, erhält automatisch dieselbe Gruppe wie das Verzeichnis selbst – egal, welcher User die Datei erstellt.

## Access Control Lists (ACL/DACL)

sind in den meisten FS-Implementierungen verfügbar; Unix-Leute benutzen sie meistens nicht!

# e.g. POSIX ACLs:

```
$ setfacl m:775,u:hans:770 file1 file2 ....  
$ setfacl -m g:programmers:rx file1 file2 ....  
$ getfacl file1 file2 ....
```

# Filesystem Kommandos

- ls
- cp
- more, pg, less
- chmod
- pwd
- mv
- pr
- chown
- cd
- ln
- file
- chgrp
- mkdir
- rm
- find
- rmdir
- cat
- touch

## Hard-Link

directory

123	name_1
123	name_5
123	name_6

## INODE speichert:

Dateityp, Zugriffsrechte (rwx), GID, UID,  
Dateigröße, Hardlinkzähler,  
Verweis auf die Datenblöcke;  
Dateiname ist NICHT enthalten!

