

1

EINFÜHRUNG

■ Was ist UNIX?



UNIX ist eine Betriebssystemfamilie



UNIX ist ein Warenzeichen



UNIX wird nach Derivaten unterschieden

■ Was ist Linux?



Linux ist Unix-ähnlich

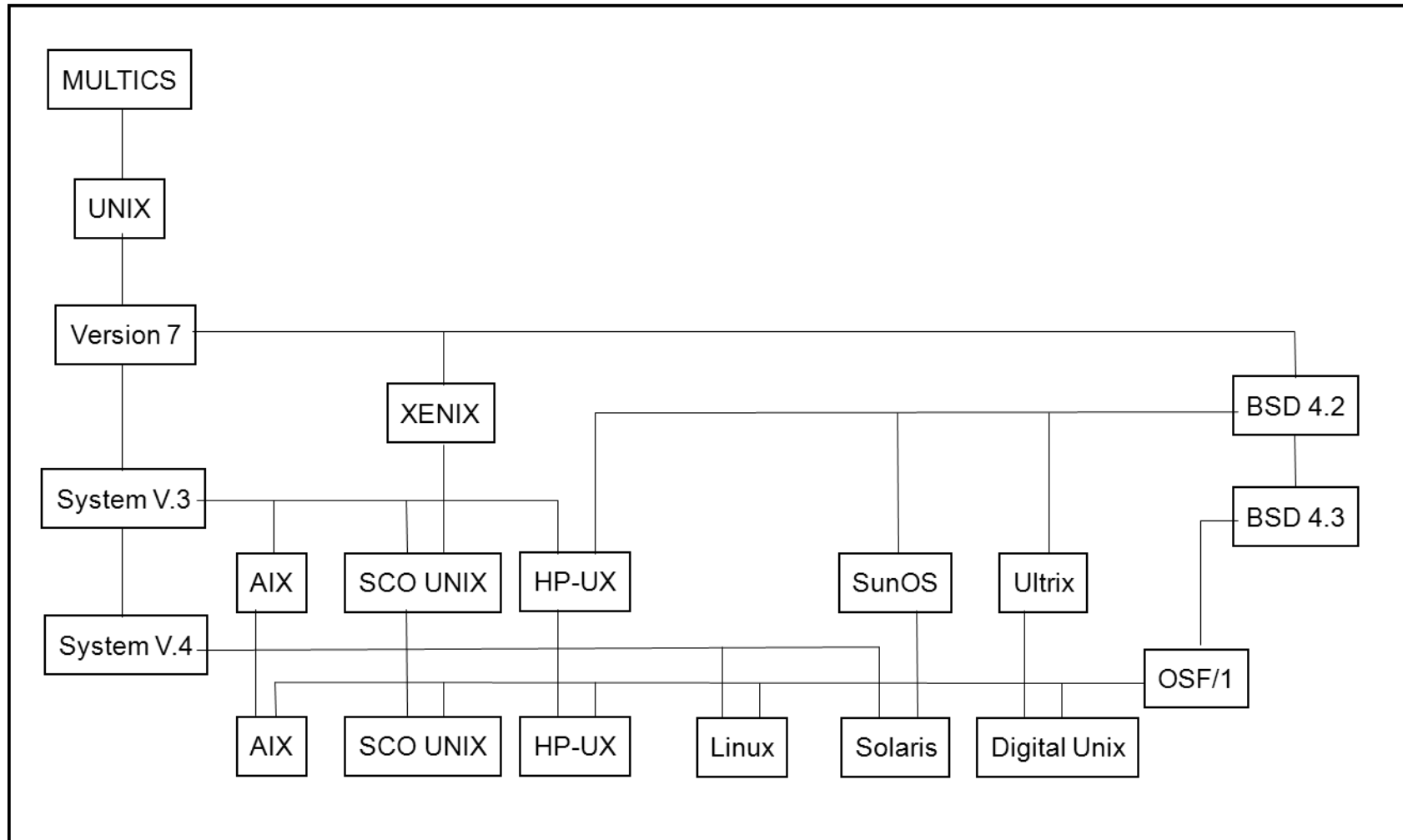


Linux ist „frei“ aber nicht immer kostenlos



Linux wird nach Distributionen unterschieden

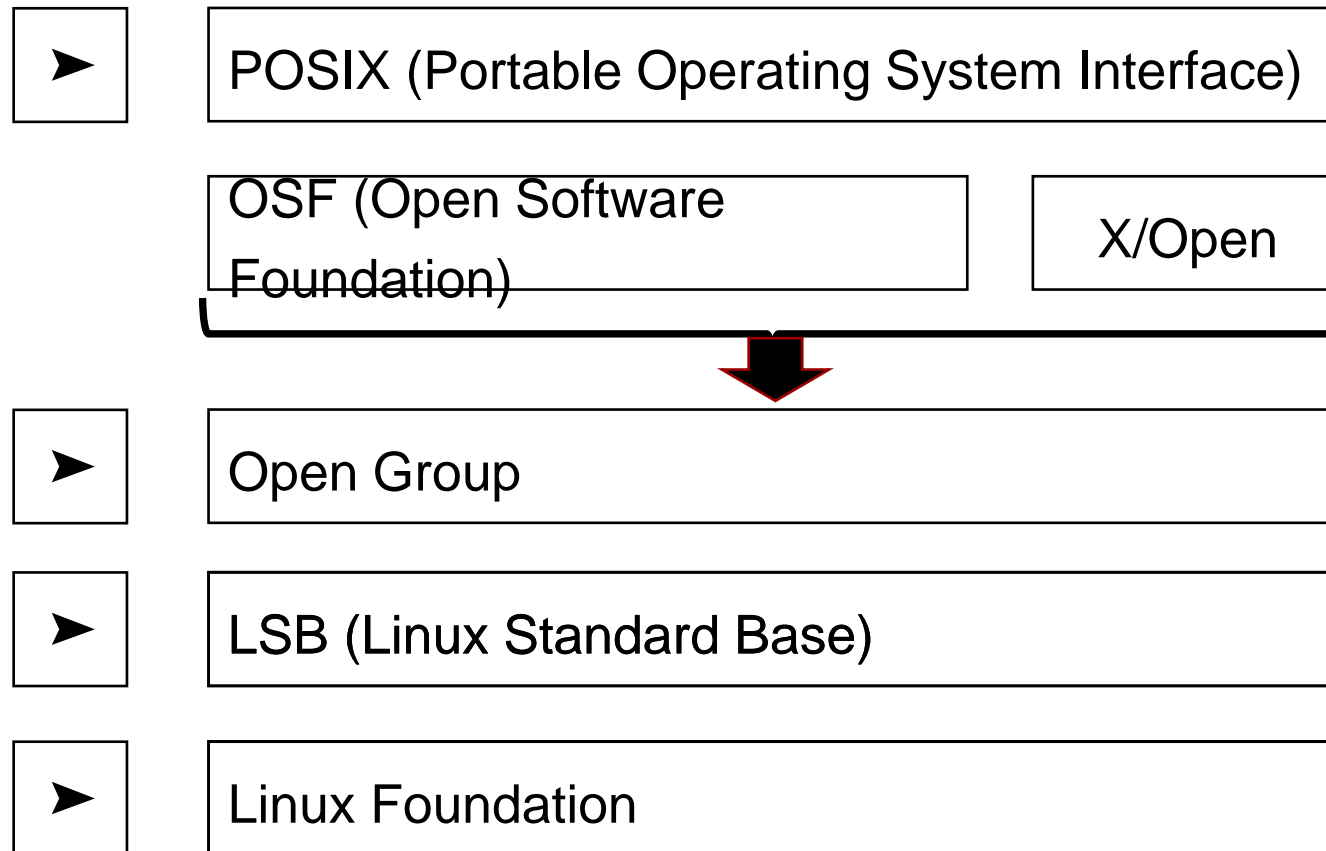
Übersicht über UNIX-Entwicklungslinien und Derivate



Die wichtigsten UNIX-Derivate / Linux Distributionen

| UNIX Derivat | Firma |
|--|---|
| SunOS / Solaris AIX HP-UX Tru64 UNIX | Oracle ab 2010 (ehem. Sun Microsystems) IBM Hewlett Packard (HP) Hewlett Packard (HP) |

| Linux Distribution | Distributor |
|--|--|
| Red Hat Enterprise (RHEL) Fedora SUSE Linux Enterprise (SLES) openSUSE Debian Ubuntu, Knoppix, Mint, Gentoo, Arch, CentOS, Mandriva, Slackware OS-X, iOS, Android, ... FreeBSD | Red Hat (USA) Fedora / Red Hat SUSE (Nürnberg DE) openSUSE-Projekt / SUSE Debian-Projekt Berkeley System Distribution |



- + Multitasking System
- + Multiuser System
- + Virtuelle Speicherverwaltung
- + Hierarchisches, geräteunabhängiges Dateisystem
- + Netzwerkfähig (TCP/IP)
- + Kleiner Betriebssystemkern
- + Bedienung per Kommandozeile oder grafischer Benutzeroberfläche

- + Vergabe von Berechtigungen für Benutzer und Gruppen, abgebildet durch Zugriffsschutz auf Dateiebene
- + Sehr viele Utilities und Dienstprogramme
- + C-Compiler mit leistungsfähigem C-Entwicklungssystem meist im Basissystem enthalten, andere Compiler sind verfügbar
- + Viele kommerzielle Anwendungen vorhanden
- + Real-Time-Fähigkeit auf allen neueren UNIX-Derivaten
- + 64-Bit-Architektur auf allen neuen Derivaten
- + Sehr stabil und auch im Hochleistungsbereich sehr zuverlässig

- Hoher Lernaufwand - vor allem bei Bedienung über Befehlszeile
(wird durch grafische Benutzeroberfläche aber abgeschwächt)
- Zu wenige, zu knappe und manchmal unklare Fehlermeldungen
(in den neueren Versionen besser)
- Bei sensiblen Befehlen in der Standardkonfiguration
teilweise keine Rückfragen, was aber konfigurierbar ist



X WINDOW SYSTEM



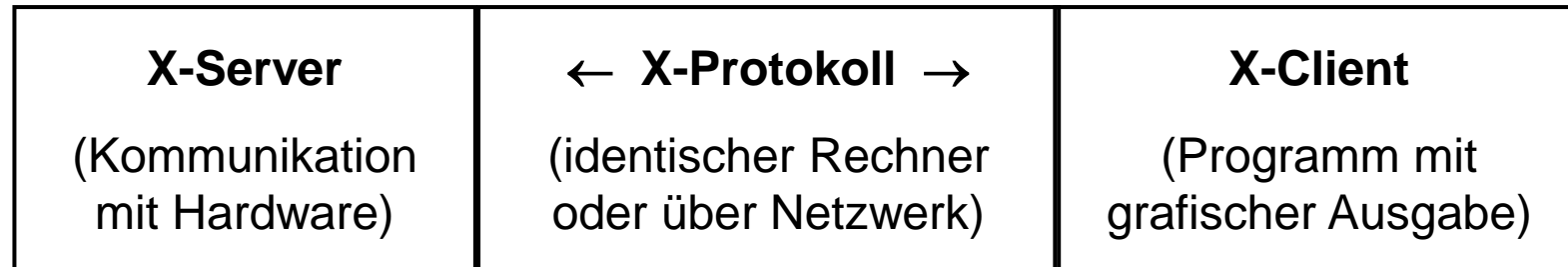
CDE-Desktop – ehemals Standardoberfläche unter Unix

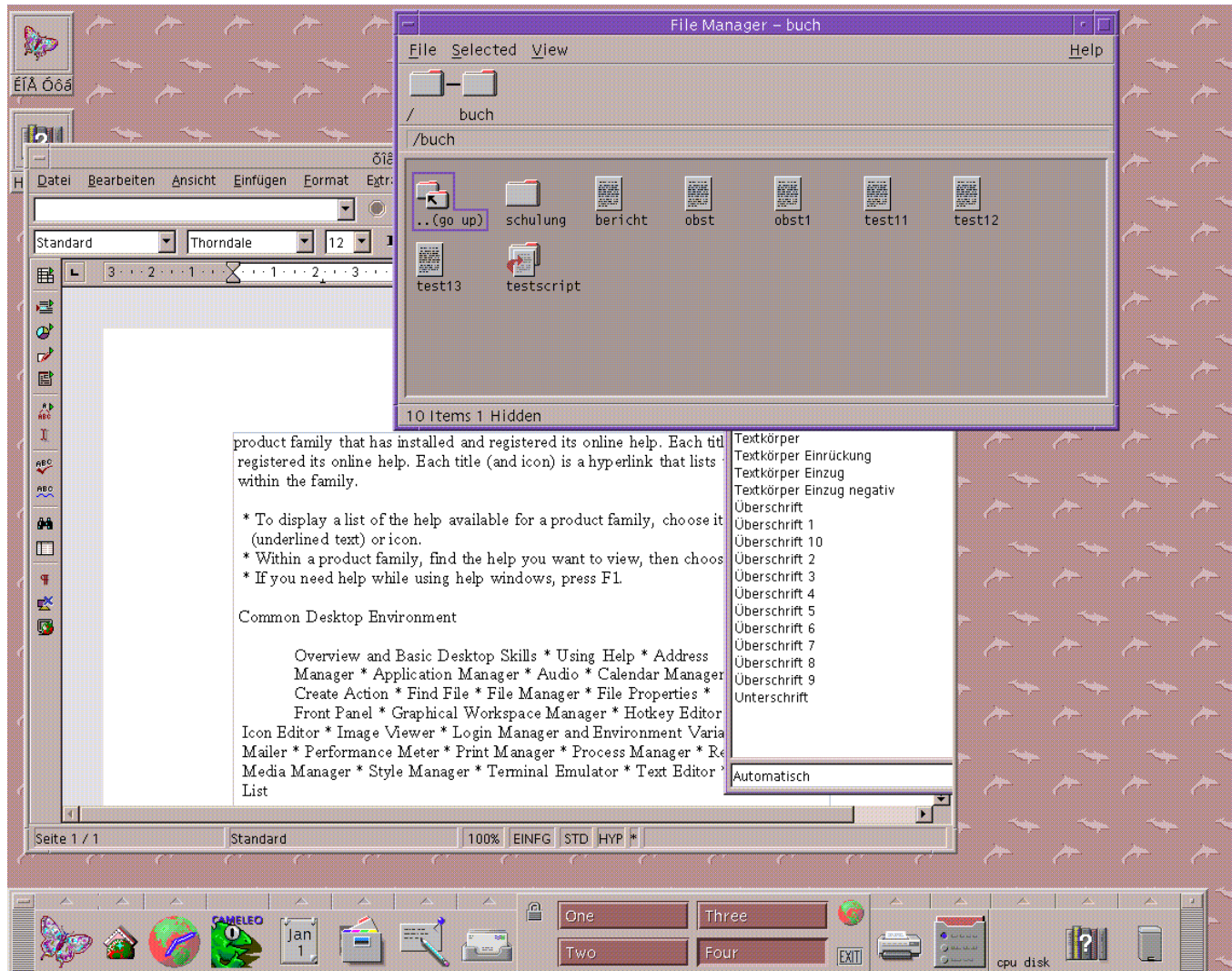


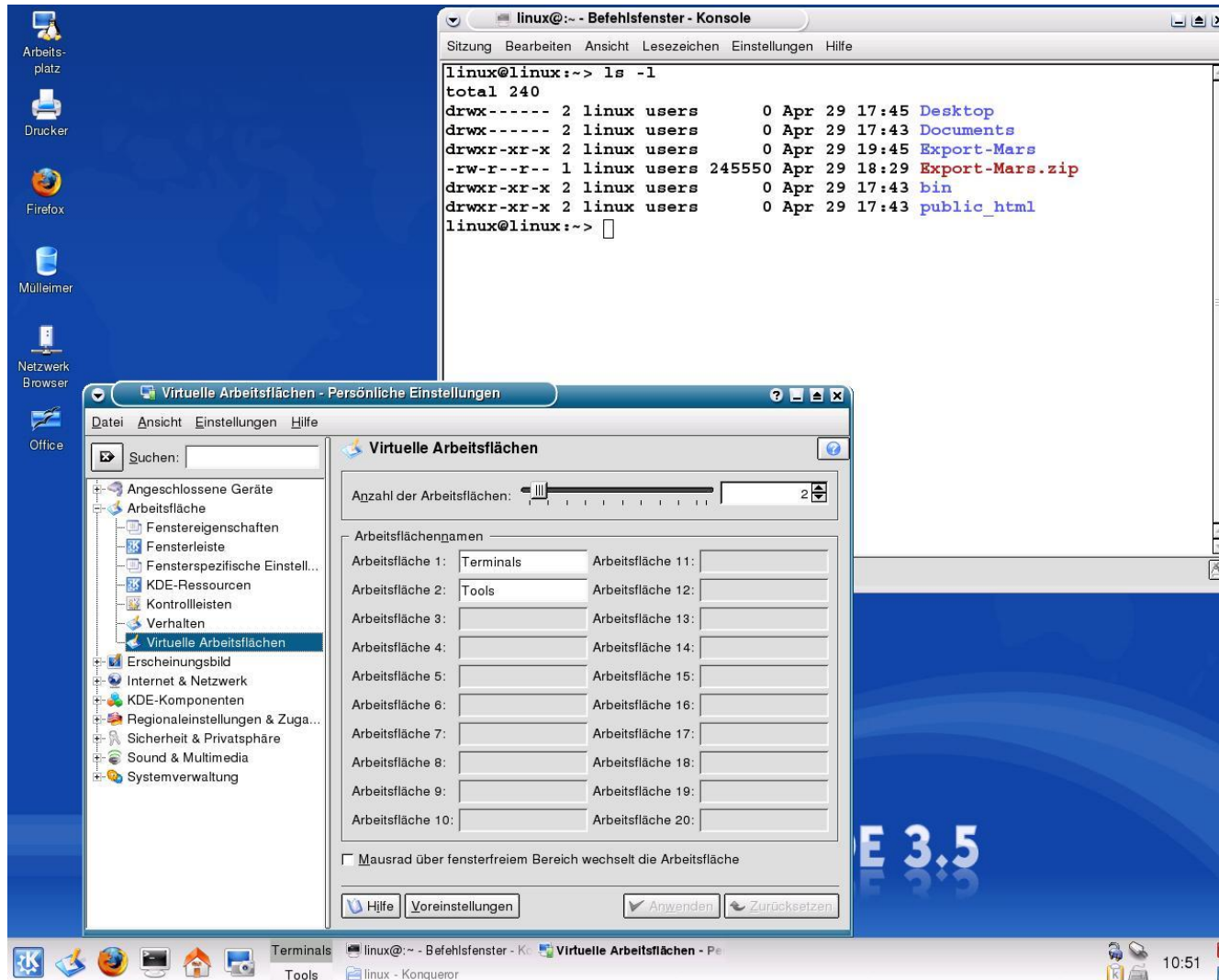
KDE-Desktop – alternative Oberfläche unter Linux

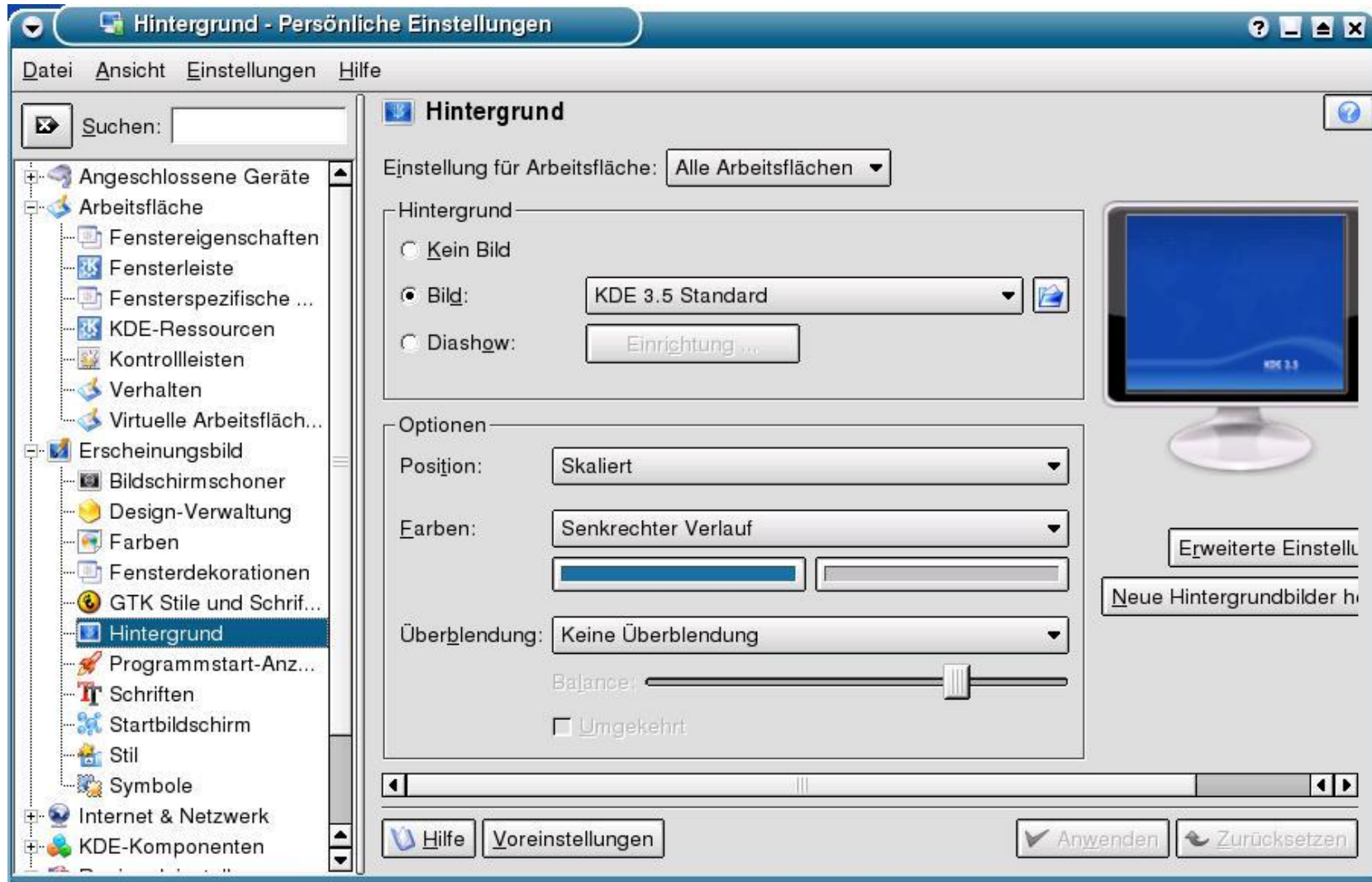


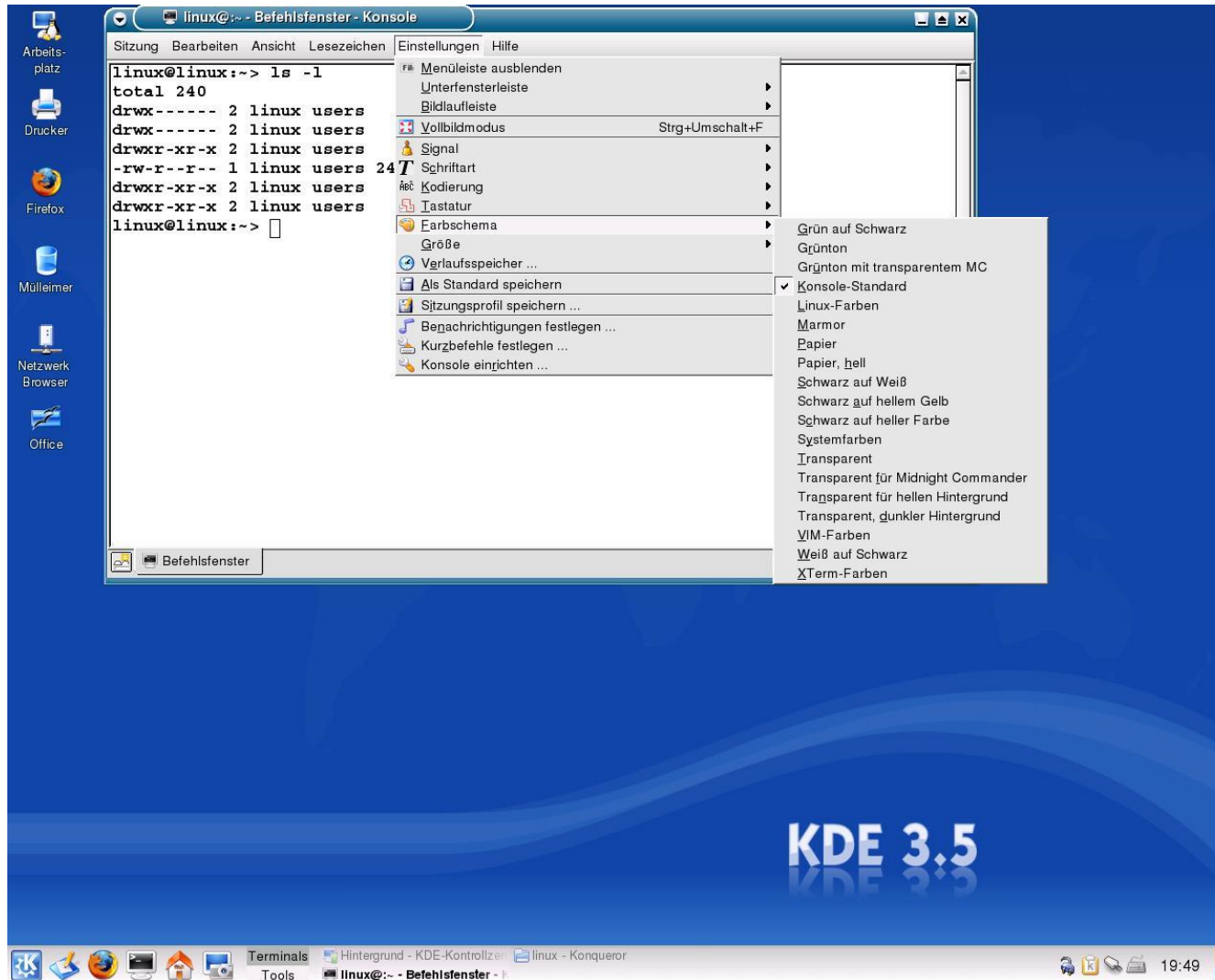
GNOME-Desktop – Standardoberfläche unter Enterprise Linux und Solaris ab V. 11











2

ERSTE SCHRITTE AM SYSTEM

Anmeldevorgang für den Benutzer sl01:

login: sl01

Password:

Last login: Wed Aug 18 10:00:45 on tty0

You have mail.

\$

Das Passwort wird nicht angezeigt, wenn es eingegeben wird.

Der Befehl passwd

Passwortänderung für den Benutzer sl01:

```
passwd
passwd: Changing passwd for sl01
Enter login password:
New passwd
Re-enter passwd:
```

Das Passwort wird auch bei einer Änderung nicht angezeigt, aber sicherheitshalber zweimal abgefragt.

befehl [*-option(en)*] [*argument1 argument2 ...*]

Beispiele:

ls

ls -l

ls -a -l

ls -la

ls /etc

ls /usr /home

ls -Rla /home/anna /home/otto

Syntax: **man** [-s *sektion*] *befehl* (Unix)
man [*sektionsnr*] *befehl* (Linux)

| 1 | Benutzerbefehle |
|---|--------------------------------------|
| 2 | Systemfunktionen (C-Funktionen) |
| 3 | Bibliotheksfunktionen (C-Funktionen) |
| 4 | Dateiformate, Gerätedateien |
| 5 | Dateiformate (Konfigurationsdateien) |
| 6 | Spiele, Demoprogramme |
| 7 | Diverses |
| 8 | Verwaltungsbefehle |

| Name | Name des Befehls |
|--------------------|---|
| Synopsis | Aufrufvarianten mit Optionen und Argumenten, die sogenannte Syntax des Befehls |
| Description | Beschreibung der Wirkungsweise des Befehls |
| Options | Verfügbare Optionen mit Argumenten und ihrer Wirkungsweise |
| Exit Code | Mögliche Rückgabewerte des Befehls, für Programmierer und Systemadministratoren interessant |
| Examples | Beispiele, wie der Befehl benutzt werden kann |
| Files | Dateien, die von diesem Befehl gelesen oder beschrieben werden |
| Bugs | Bekannte Probleme, Bugs oder systemspezifische Einschränkungen, sofern bekannt |
| See Also | Hinweis auf ähnliche Befehle, Dateiformate, etc. |

- Kurzinformation abrufen

Syntax: **man -f** *befehl*
whatis *befehl*

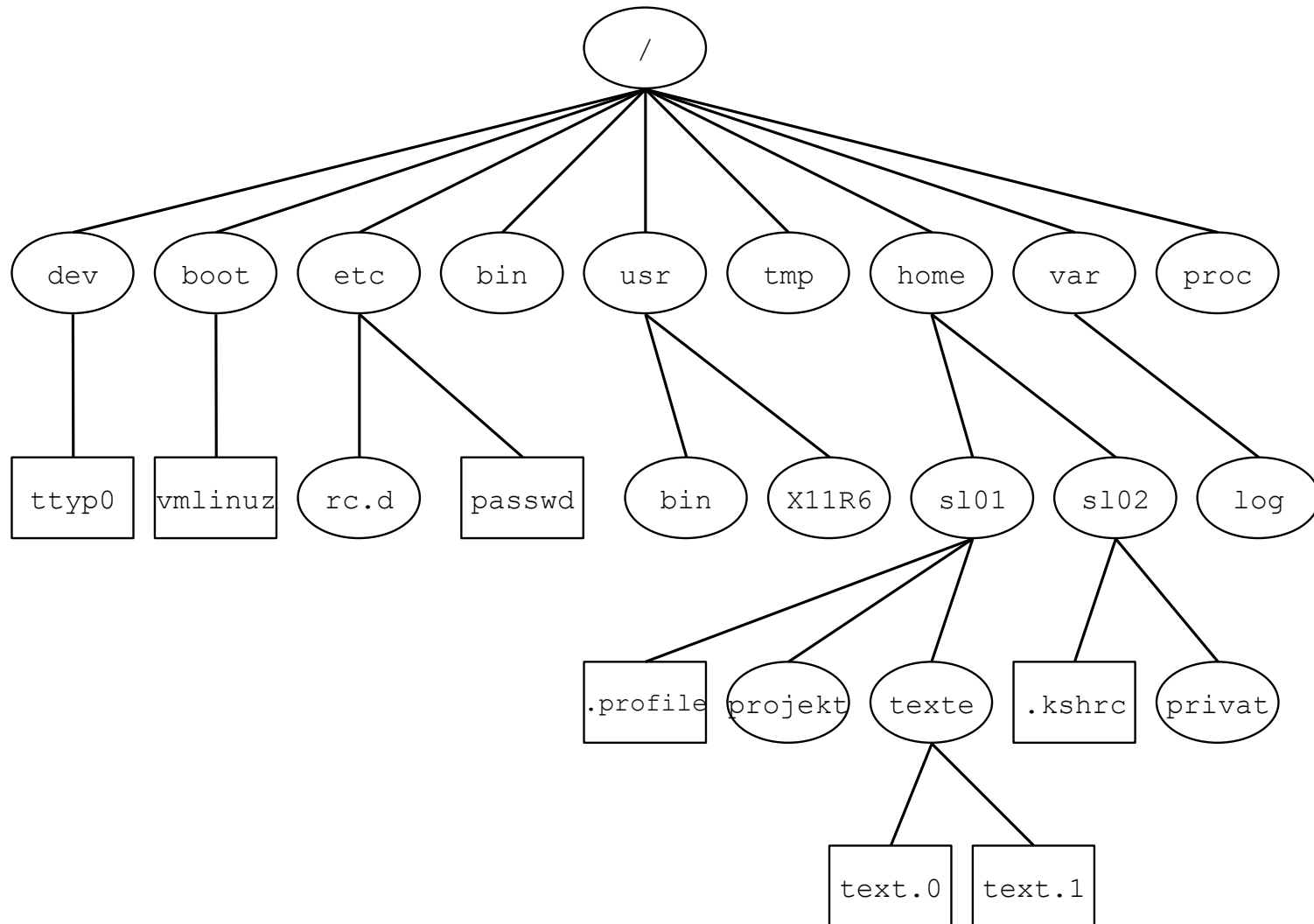
- Schlüsselwortsuche

Syntax: **man -k** *schlüsselwort*
apropos *schlüsselwort*

- Texte am Bildschirm ausgeben - echo
- Anmeldenamen anzeigen - logname
- Angemeldete Benutzer anzeigen - who
- Informationen zu Benutzern anzeigen - finger
- Datum und Uhrzeit anzeigen - date

3

DATEIVERWALTUNG



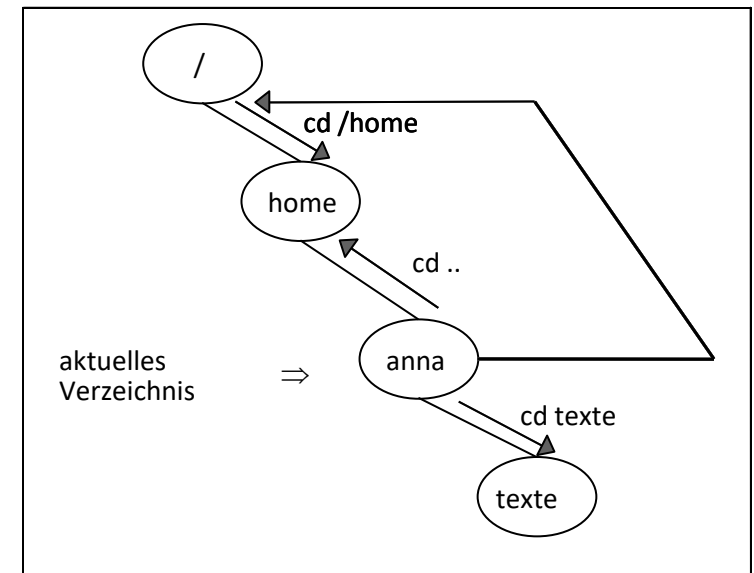
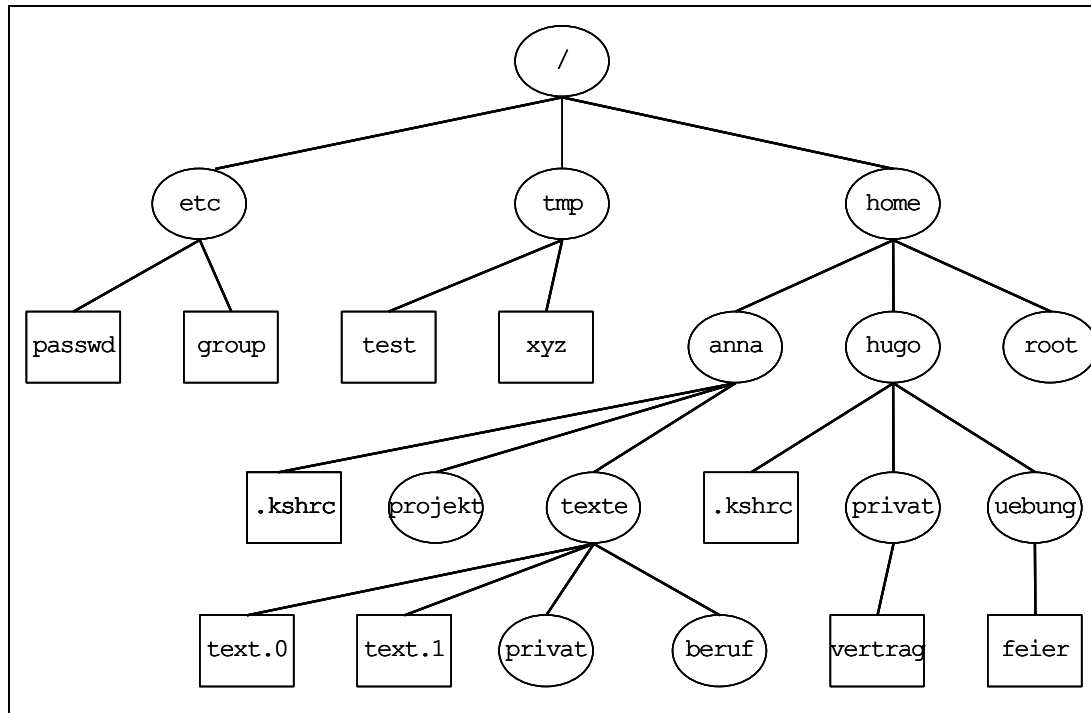
- root-Verzeichnis
- Homeverzeichnis
- Arbeitsverzeichnis
- Übergeordnetes Verzeichnis

- **/bin, /usr/bin** Normale Benutzerbefehle
- **/opt** Optionale Software
- **/usr** Unix System Resources
- **/tmp, /var/tmp** temporäre Dateien

- **/lib, /usr/lib** Bibliotheksroutinen
- **/etc** Verwaltungs- und Konfigurationsdateien
- **/proc** Prozess-Verwaltung
- **/sbin, /usr/sbin** Befehle für die Systemverwaltung

- **/home** Homeverzeichnisse der Benutzer (Linux)
- **/export/home** Homeverzeichnisse der Benutzer (Solaris)

- Reguläre Dateien
- Verzeichnisse
- Symbolische Links
- Spezialdateien
 - Blockorientierte Gerätedateien
 - Zeichenorientierte Gerätedateien
 - Named Pipes
 - Sockets
 - Doors



- ☐ Aktuelles Verzeichnis anzeigen - **pwd**
- ☐ Verzeichnisinhalt listen - **ls**
- ☐ Verzeichnis wechseln - **cd**
- ☐ Verzeichnis anlegen - **mkdir**
- ☐ Leeres Verzeichnis löschen - **rmdir**

- ☐ Zeitstempel ändern/Leere Dateien anlegen - **touch**
- ☐ Dateien/Verzeichnisse kopieren - **cp**
- ☐ Dateien/Verzeichnisse umbenennen/verschieben - **mv**
- ☐ Dateien/Verzeichnisse löschen - **rm**

| -i | interactive: Nachfrage, ob die Aktion durchgeführt werden soll, wenn dadurch eine bestehende Datei überschrieben werden würde |
|-----------|---|
| -r | recursive: Rekursives Kopieren ganzer Verzeichnisbäume (Verzeichnis mit allen darin enthaltenen Dateien und Unterverzeichnissen) |
| -f | force: Aktion in jedem Fall durchführen (ohne Rückfrage) |

- Dateiinhalt ungepuffert ausgeben - **cat**
- Dateiinhalt seitenweise ausgeben - **more**
- Dateiinhalt seitenweise ausgeben - **pg**
- Lokalisierung über die Shell - **type**
- Lokalisierung im System - **whereis**
- Dateityp bestimmen - **file**
- Links anlegen - **ln**

4

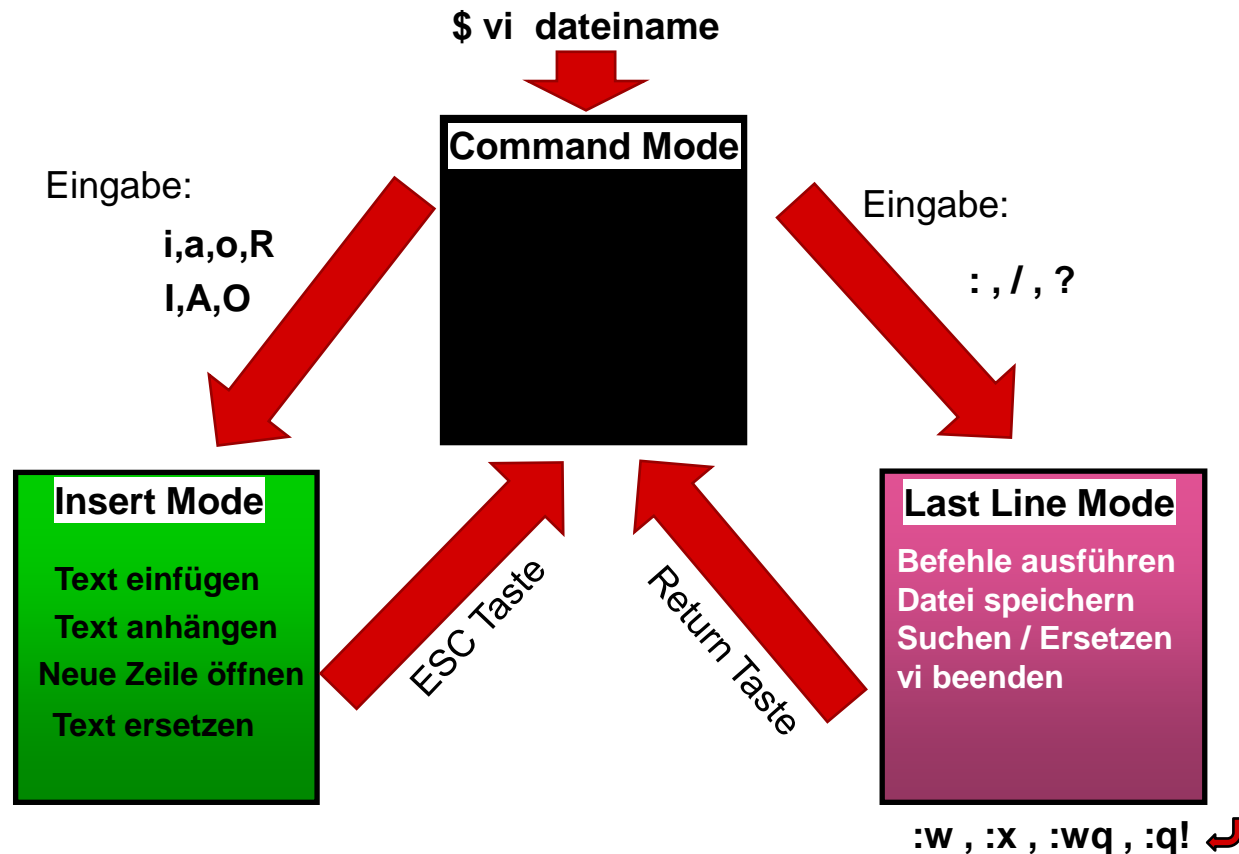
EDITOREN

□ Interaktive Editoren:

- Zeilenorientierter Editor ed bzw. ex (Standard)
- Bildschirmorientierter Editor vi (Standard)
- Bildschirmorientierter Editor emacs
(Erweiterung)

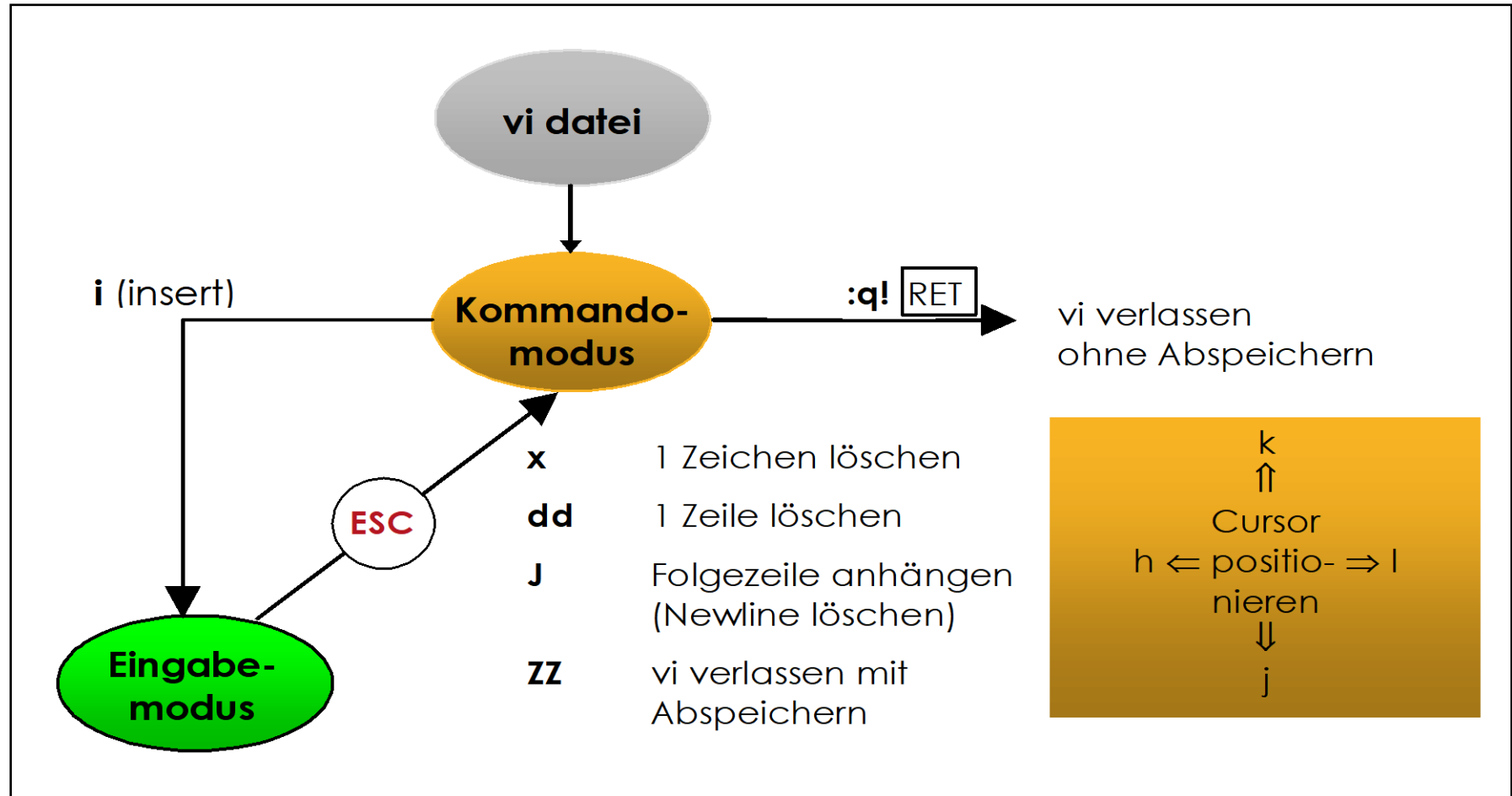
□ Nicht interaktiver Editor:

- Streameditor sed (Standard)



| Kommando | Bedeutung |
|-------------------|--|
| vi [datei] | Aufruf des vi |
| h ⇐ | Cursor nach links |
| l ⇒ | Cursor nach rechts |
| k ↑ | Cursor nach oben |
| j ↓ | Cursor nach unten |
| x | Zeichen an Cursorposition löschen |
| dd | aktuelle Zeile löschen (delete) |
| J | Folgezeile an aktuelle Zeile anhängen (join) |
| i | Wechsel in den Eingabemodus an der aktuellen Cursorposition (insert) |
| a | Wechsel in den Eingabemodus hinter der aktuellen Cursorposition (append) |
| ESC | Text-Modus beenden |
| ZZ | vi mit Abspeichern verlassen |
| :q! | vi ohne Abspeichern verlassen |

Die wichtigsten vi-Befehle in grafischer Übersicht



- **i** Einfügen von Text an der aktuellen Cursorposition
- **a** Einfügen von Text rechts vom Cursor
- **I** Einfügen von Text am Zeilenanfang
- **A** Anhängen von Text am Zeilenende
- **O** Öffnet eine Leerzeile vor der aktuellen Zeile
- **o** Öffnet eine Leerzeile nach der aktuellen Zeile
- **R** Überschreibt den Text in der aktuellen Zeile
- **r** Ersetzt das Zeichen an Cursorposition

- **h** ⇐ 1 Position nach links
- **l** ⇒ 1 Position nach rechts
- **k, -** ↑↑ 1 Zeile nach oben
- **j, +** ↓↓ 1 Zeile nach unten
- **0** an den Zeilenanfang (null)
- **^** auf das erste sichtbare Zeichen
- **\$** an das Zeilenende

- **w** zum nächsten Wort
- **b** zum vorhergehenden Wort
- **nG** zur Zeile *n*
- **G** zur letzten Zeile
- **^F** eine Seite vorwärts Blättern
- **^B** eine Seite rückwärts Blättern
- **fx** nach rechts zum 1. Auftreten des Zeichens *x*

- **x** löscht Zeichen an aktueller Cursorposition
- **X** löscht Zeichen links vom Cursor
- **ndw** löscht ein / *n* Worte ab Cursorposition
- **d\$** löscht von der Cursorposition bis zum Zeilenende
- **ndd** löscht *n* Zeilen ab der aktuellen Zeile
- **nyy** aktuelle bzw. *n* Zeilen in den Puffer kopieren
- **P** Inhalt des Puffers *vor* aktueller Cursorposition bzw. Zeile einfügen
- **p** Inhalt des Puffers *hinter* aktueller Cursorposition bzw. Zeile einfügen
- **J** An die aktuelle Zeile wird die nachfolgende gehängt
- **u** letzte Textänderung rückgängig machen

- **/***pattern* Suchen vorwärts nach dem Textmuster *abc*
- **?***pattern* Suchen rückwärts nach dem Textmuster *abc*
- **n** Wiederholung der letzten Suche
- **N** Wiederholung der letzten Suche in umgekehrter Richtung

- ***n*sText** *n* Zeichen werden durch Text ersetzt
- ***n*cwText** Ersetzt ein bzw. *n* Worte ab Cursorposition durch Text
- **ct**xText Ersetzt den Text bis zum nächsten beliebigen Buchstaben *x*
- **c\$**Text Ersetzt den Text bis zum Zeilenende
- ***n*ccText** *n* Zeilen werden durch Text ersetzt
- **.** Wiederholung des letzten Befehls

[adr1[,adr2]]s/suchmuster/ersatztext/Flag

| Adressangabe | Bedeutung |
|--|---|
| keine Angabe | Aktuelle Zeile |
| <i>adr1</i> | Zeile mit Adresse <i>adr1</i> |
| <i>adr1,adr2</i> | von Zeile mit Adresse <i>adr1</i> bis Zeile mit Adresse <i>adr2</i> |
| Adresse kann sein | Bedeutung |
| . | aktuelle Zeile (nur sinnvoll in Kombination mit einer 2. Adressangabe, bspw. 1,.) |
| \$ | letzte Zeile |
| % | alle Zeilen (entspr. 1,\$) |
| <i>n</i> | Zeile Nr. <i>n</i> |
| <i>+n</i> oder <i>-n</i> | relative Adressierung bezogen auf die aktuelle Zeile (<i>+n</i> – <i>n</i> Zeilen hinter / <i>-n</i> – <i>n</i> Zeilen vor der Aktuellen) |
| <i>/pattern/</i> | Zeile die <i>pattern</i> enthält |

➤ von,bis**s**/suchmuster/ersatztext/

:2,15**s**/*abc*/*xyz*/**g**

Von Zeile 2 bis 15 werden **alle** Muster *abc* durch *xyz* ersetzt (**g** = **g**lobal)

➤ von,bis**s**/suchmuster/ersatztext/**g**

:-5,+5**s**/*abc*/*xyz*/**g**

Im Bereich von 5 Zeilen vor bis 5 Zeilen hinter dem Cursor wird das Muster *abc* durch *xyz* ersetzt

➤ von,bis**s**/suchmuster/ersatztext/**gc**

::\$**s**/*abc*/*xyz*/**gc**

Von der aktuellen (.) bis zur letzten (\$) Zeile wird das Muster *abc* durch *xyz* im interaktiven Modus ersetzt

- **ZZ** Sichern und beenden (schnellste Variante)
- **:x** Sichern und beenden (im Befehlszeilenmodus)
- **:wq** Sichern und beenden (im Befehlszeilenmodus)
- **:q** Beenden ohne Abspeicherung (Fehlermeldung, falls die Datei verändert wurde)
- **:q!** Beenden ohne Abspeicherung (ohne Nachfrage)
- **:w [datei]** Sichern ohne Beenden (ggf. in angegebene Datei)

- **CTRL-g** Liefert Informationen über die aktuelle Datei
- **CTRL-I** Neuaufbau des Bildschirms
- ***n*>>** Am Anfang der nächsten *n* Zeilen wird ein Tabulator eingefügt
- ***n*<<** Am Anfang der nächsten *n* Zeilen wird ein Tabulator entfernt
- **%** Steht der Cursor auf einer Klammer ([{ }]), so liefert % die dazu gehörende öffnende oder schließende Klammer
- **~** Ändert Großbuchstaben in kleine und umgekehrt (z.B. 7~ ändert in den nächsten 7 Zeichen Groß- in Kleinschreibung und umgekehrt)

- `:12,17w datei` Der Zeilenbereich 12 bis 17 wird in die angegebene Datei geschrieben
 - `:r datei` Der Inhalt der Datei wird nach der aktuellen Zeile eingelesen
-
- `:r !befehl` Der UNIX-Befehl *befehl* wird ausgeführt und sein Ergebnis hinter der aktuellen Zeile abgelegt (analog `!!befehl`)
 - `!!befehl` Der UNIX-Befehl *befehl* wird ausgeführt und sein Ergebnis hinter der aktuellen Zeile abgelegt
 - `:! befehl` Der UNIX-Befehl *befehl* wird ausgeführt, der Editor nicht verlassen
 - `!!!` Wiederholt den letzten UNIX-Befehl

- **:set all** Ansehen aller Voreinstellmöglichkeiten
- **:set** Ansehen der veränderten Voreinstellungen
- **:set option** Ändern einer Voreinstellung (Option festlegen)
- **:set nooption** Ändern einer Voreinstellung (Option aufheben)

- **:set showmode** Anzeige des Einfüge- oder Anhängmodus
- **:set [no]number** Anzeige der Zeilennummerierung ein/ausschalten
- **:set autoindent** Automatisches Einrücken
- **:set showmatch** Im Einfügemodus wird beim Eingeben einer Klammer **) ,] , }** der Cursor kurz auf die dazugehörige öffnende Klammer positioniert
- **:set ignorecase** Beim Suchen nach Zeichenketten wird nicht zwischen
- **:set tabstop=4** Klein- und Großbuchstaben unterschieden
Tabulator-Schrittweite auf 4 setzen
- **:set shiftwidth=4** Schrittweite für die Einrückung mit << bzw. >> auf 4 Zeichen festlegen (Empfehlung: shiftwidth=tabstop)

5

KORN-SHELL UND BOURNE-AGAIN-SHELL

□ Die Shell erfüllt zwei Aufgaben im UNIX-System:

- Interaktiver Kommandointerpreter
- Programmiersprache

□ Die Shell lässt sich folgendermaßen beschreiben:

- Die Shell ist die Schnittstelle zwischen Anwender und Betriebssystem
- Die Shell führt Befehle aus
- Die Shell stellt Variablen bereit und besitzt Grundstrukturen der Ablaufkontrolle
- Die Shell ist ein eigener Prozess

| Shell | Kennzeichen |
|----------------------------------|---|
| Bourne-Shell (sh) | Älteste Shell; System-Shellskripte sollten soweit möglich immer kompatibel zur Bourne-Shell-Syntax geschrieben werden, da diese der kleinste gemeinsame Nenner aller Shells ist. |
| C-Shell (csh) | An der Berkeley University entwickelt, angelehnt in der Syntax an die Programmiersprache C; bedeutend komfortabler und mächtiger als die Bourne-Shell durch viele Erweiterungen. |
| Korn-Shell (ksh) | Erweiterte und zur Bourne-Shell kompatible Shell; sie hat sich als Quasistandard unter UNIX durchgesetzt. |
| Bourne-Again-Shell (bash) | Verbesserte Bourne-Shell mit vielen Eigenschaften der C- und Korn-Shell; sie ist die Standardshell unter Linux. |
| Tenix C-Shell (tcsh) | Deutlich verbesserte C-Shell mit Korn-Shell-Erweiterungen und neuen Zusatzeigenschaften. |
| Z-Shell (zsh) | Eine moderne und komfortable Shell, die sowohl Skripte in Bourne-Shell- als auch in C-Shell-Sprache versteht und deutliche Verbesserungen der Bourne-Again- und der TC-Shell enthält. |

- **history** [anzahl]
- **r** [kommandoname / kommandonr]

Beispielaufufe:

| Befehl | Aktion |
|------------------------|---|
| history | Ausgabe der letzten 16 Befehle mit Nummern |
| history -n | Ausgabe der letzten <i>n</i> Befehle mit Nummern |
| history n m | Ausgabe der Kommandozeilen von Nummer <i>n</i> bis <i>m</i> |
| r | Erneutes Starten des zuletzt abgesetzten Befehls |
| r -n | Erneutes Starten des Befehls, der <i>n</i> Kommandozeilen zurück liegt |
| r n | Erneutes Starten des Befehls mit der Nummer <i>n</i> |
| r string | Erneutes Starten des letzten Befehls, der mit <i>string</i> begann |
| r dat1=dat2 str | Erneutes Starten des letzten Befehls, der mit <i>str</i> begann, wobei vorher noch <i>dat1</i> durch <i>dat2</i> ersetzt wird |



set -o vi

| Befehl | Aktion |
|------------------------|--|
| ESC | Wechsel in den Befehlsmodus |
| k oder - | Cursor nach oben (History: vorherigen Befehl holen) |
| j oder + | Cursor nach unten (History: nächsten Befehl holen) |
| h | Cursor nach links |
| l | Cursor nach rechts |
| x | Zeichen unterhalb des Cursors löschen |
| i | Wechsel in den Eingabemodus, um Zeichen links von der Cursorposition einzufügen (insert) |
| a | Wechsel in Eingabemodus, um Zeichen rechts von der Cursorposition einzufügen (append) |
| cw | Wechsel in Eingabemodus und den Text ab Cursorposition bis Wortende durch nachfolgenden Text ersetzen (change word) |
| /muster | Nach Befehl mit angegeben <i>muster</i> in bereits eingegebenen Befehlszeilen suchen (wiederholtes Suchen mit n) |
| ESC \ | Dateinamen vervollständigen |

□ set -o emacs

alias __A=^P

alias __B=^N

alias __C=^F

alias __D=^B

| Befehl | Aktion |
|---------|---|
| ↑ | Cursor nach oben (History: vorherigen Befehl holen) |
| ↓ | Cursor nach unten (History: nächsten Befehl holen) |
| ← | Cursor nach links bewegen |
| → | Cursor nach rechts bewegen |
| ⌫ | Zeichen links vom Cursor löschen |
| CTRL d | Zeichen unterhalb des Cursors löschen |
| Ctrl a | Cursor an den Zeilenanfang |
| Ctrl e | Cursor an das Zeilenende |
| ESC ESC | Dateinamen vervollständigen |

Syntax: **alias** [-optionen] [*name*[=*kommando*]]

| Kommando | Bedeutung |
|--|---|
| alias | Alle Alias-Definitionen anzeigen |
| alias <i>name</i> | Alias-Definition für <i>name</i> anzeigen |
| alias <i>name</i> = <i>kommando</i> | Alias definieren |
| unalias <i>name</i> | Alias-Definition für <i>name</i> löschen |

- Unterscheidung nach Herkunft:
 - Systemvariable
 - Benutzerdefinierte Variable

- Unterscheidung nach Sichtbarkeit:
 - Lokale Variable
 - Umgebungsvariable

- Anzeige von Variablen:
 - Alle anzeigen mit dem Befehl set
 - Umgebungsvariable anzeigen mit dem Befehl env
 - Inhalt anzeigen mit echo \$var

- Setzen von Variablen:
 - Lokale Variable mit var=wert
 - Umgebungsvariable mit export var

| Variable | Bedeutung |
|-----------------|--|
| HOME | Absoluter Pfad zum Homeverzeichnis |
| PATH | PATH enthält die Verzeichnisse, die die Shell durchsucht, um einen Befehl zu finden. |
| EDITOR | Legt den Standard-Editor fest |
| PS1 | Promptzeichen der Shell (Shellvariable) |
| PS2 | Folgepromptzeichen, falls ein Befehl in einer Zeile nicht abgeschlossen ist. (Shellvariable) |
| PWD | Enthält den Pfad für das jeweils aktuelle Verzeichnis |
| TERM | Terminaltyp (wichtig für Editoren) |
| LOGNAME | Anmeldename eines Benutzers |
| USER | Aktuelle Benutzerkennung |
| SHELL | Anmeldeshell eines Benutzers |
| HOSTNAME | Rechnername, auf dem der Benutzer arbeitet |
| OSTYPE | Verwendetes Betriebssystem |

| Wildcard | Bedeutung |
|------------------|--|
| * | kein, ein oder mehrere beliebige Zeichen |
| ? | genau ein beliebiges Zeichen |
| [abc] oder [a-c] | ein Zeichen aus der Menge <i>abc</i> |
| [!abc] | ein Zeichen, jedoch nicht <i>a</i> oder <i>b</i> oder <i>c</i> |
| ~ | absoluter Pfad zum eigenen Home-Verzeichnis |
| ~user | absoluter Pfad zum Login-Verzeichnis des angegebenen Users |
| ~- | absoluter Pfad zum vorherigen Verzeichnis (aus dem man zuletzt mit <i>cd</i> gewechselt ist) |

- Die nachfolgende Tabelle zeigt die 3 Möglichkeiten für das Quotieren:

| Quoting | | | | | | |
|---------|---|---|---|---|---|---|
| \ | - | - | - | - | - | - |
| ' ... ' | - | - | - | x | - | v |
| " ... " | + | - | + | - | x | v |

Legende:

| | |
|---|--|
| - | Sonderbedeutung aufgehoben |
| + | Sonderbedeutung bleibt erhalten |
| v | Sonderbedeutung „Kmd.- Zeilenende“ aufgehoben, Sonderbedeutung „Neue Zeile“ bleibt erhalten |
| x | beendet entsprechendes Quoting |

□ Ersetzung eines Befehls während der Abarbeitung durch seine Ausgabe

□ Syntax:

``befehl ...`` (in sh, ksh, bash)

`$(befehl ...)` (in ksh und bash)

□ Beispiel:

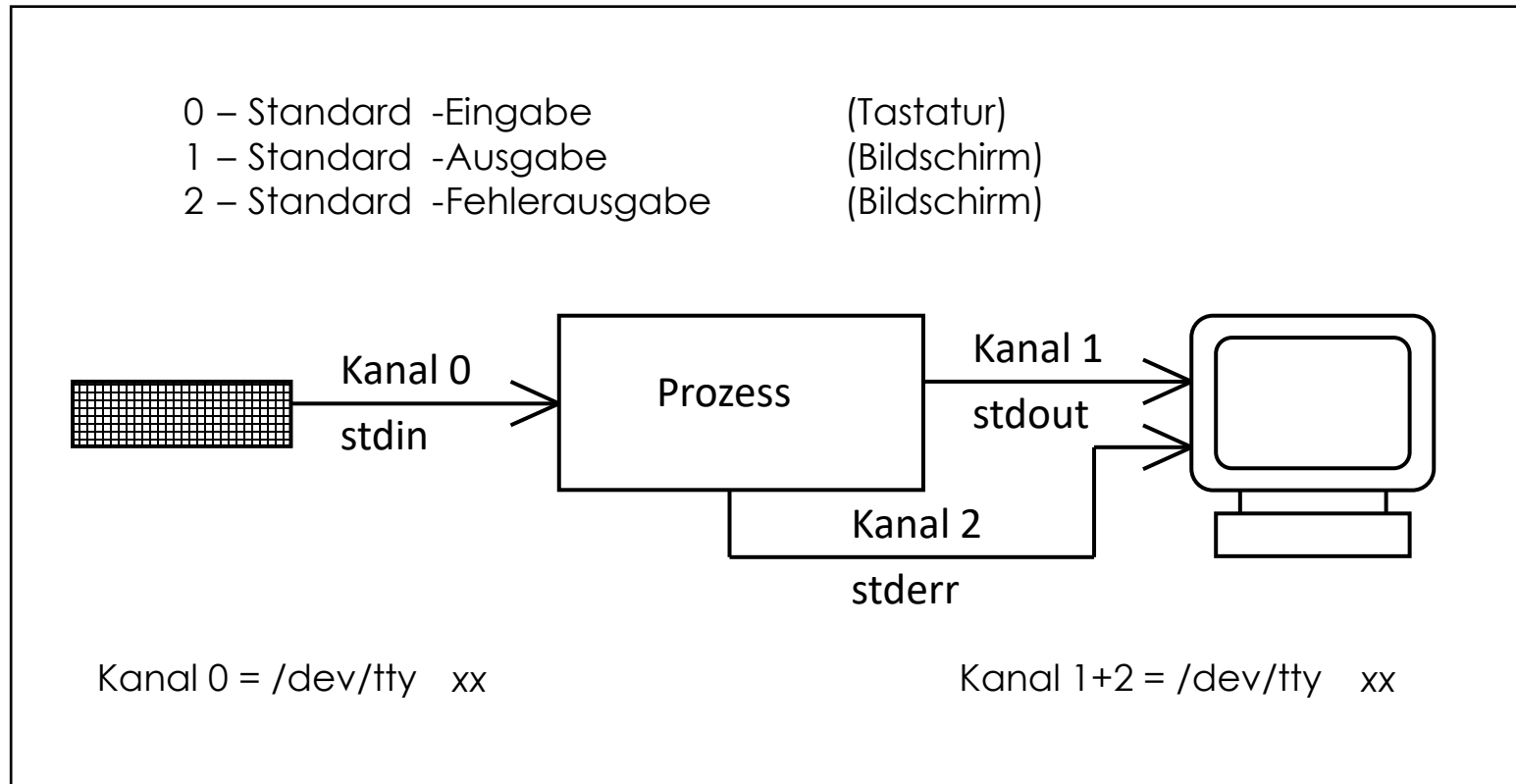
```
$ zeit=$(date +%T)
```

```
$ echo "Aktuelles Datum: $(date +%d.%m.%Y)\nAktuelle Zeit: $zeit"
```

```
Aktuelles Datum: 30.03.2009
```

```
Aktuelle Zeit: 10:15:40
```

Umlenkung der Ein- und Ausgabe



□ Umlenkung der Standardeingabe (stdin)

befehl < datei

Beispiel: **mail** *user* < *anhang*

□ Umlenkung der Standardausgabe (stdout)

befehl > datei

Beispiel: **ls -l** > *liste*

Anhängen an eine Datei: *befehl >> datei*

□ Umlenkung der Standardfehlerausgabe (stderr)

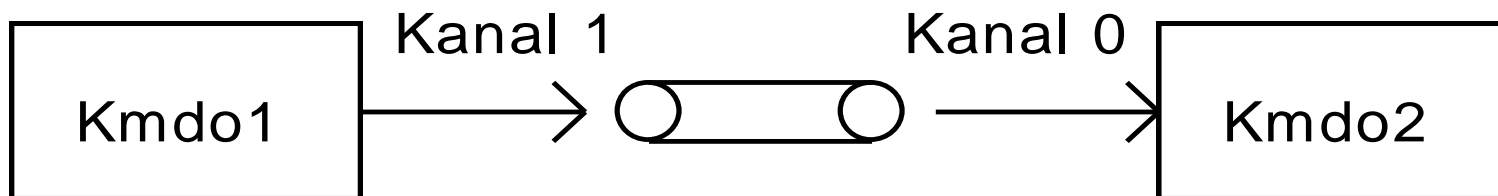
befehl 2> datei

Beispiel: **ls /etc /no** 2> *fehler*

Beim Pipe-Mechanismus bildet die Ausgabe eines Befehls die Eingabe eines anderen Befehls.

kommando1 | kommando2 | kommando3 ...

Beispiel: **ls | wc -l**
112



□ Befehle listen

befehl ; befehl ; befehl ...

Beispiel: **ls -l ; cd /etc ; pwd**

□ Kommando-Gruppierung

➤ mit eigenem Subprozess

(*kmd1; kmd2; ...*)

➤ Reine Gruppierung

{ *kmd1; kmd2; ...* ; }

- Optionen steuern das Verhalten der Shell
- Anzeige von Optionen mit dem Befehl **set**
- Aktivieren einer Option

set -o option

Beispiel: **set -o noclobber**

- Deaktivieren einer Option

set +o option

Beispiel: **set +o noclobber**

- Die nachfolgende Tabelle zeigt die wichtigsten Schalter für die interaktive Arbeit:

| Schalter (Default/Empfehlung) | | Bedeutung |
|--|-------------------|---|
| bgnice | (on/on) | Hintergrund-Jobs mit schlechterer Priorität ausführen |
| vi | (off/ on) | Aktivieren des Built-in-Editors vi für Kmd.-Zeilen-History (vergleiche Abschnitt 5.2.1.2) |
| emacs | (off/ on) | Aktivieren des Built-in-Editors vi für Kmd.-Zeilen-History (vergleiche Abschnitt 5.2.1.3) |
| ignoreeof (off/ on) | | Login-Shell kann nicht mit ^D beendet werden; Logout muss über exit erfolgen |
| noclobber | (off/ on) | Existierende Datei kann nicht mit Ausgabe-Umlenkung überschrieben werden (vergleiche Abschnitt 5.7.3) |
| markdirs) | (off/-) | Bei der Dateinamen-Expansion alle Directories mit einem / ergänzen |

- Überblick über die verschiedenen Konfigurationsdateien der unterschiedlichen Shell

| Datei | sh | ksh | bash | csH |
|-----------------|----|----------|------------------|-----|
| /etc/profile | x | x | x | |
| /etc/.login | | | | x |
| ~/.profile | x | x | (x) ² | |
| ~/.kshrc (*) | | x | | |
| ~/.bash_profile | | | (x) ³ | |
| ~/.bashrc | | | x | |
| ~/.bash_logout | | | (x) ³ | |
| ~/.login | | | | x |
| ~/.cshrc | | | | x |
| ~/.logout | | | | x |

*nur falls Variable ENV=~/.kshrc gesetzt wurde

2 Distributionsabhängig

3 optional

6

WICHTIGE DIENSTPROGRAMME

| Kommando | Bedeutung |
|--------------|---|
| fgrep | Einfache Suchtextbeschreibung in Form von String-Konstanten (<i>fast grep</i>), am schnellsten |
| grep | Unterstützt Sonderzeichen für die Beschreibung von regulären Ausdrücken (BRE's) an der Stelle des Suchtextes |
| egrep | Unterstützt Sonderzeichen für die Beschreibung von regulären Ausdrücken (ERE's) an der Stelle des Suchtextes, mehrere Musterbeschreibungen können mit einem logischen ODER verkettet werden (<i>extended grep</i>) |

Syntax: `grep [optionen] muster [datei(en)]`

| Option | Bedeutung |
|------------------|---|
| -c | nur die Anzahl der gefundenen Zeilen ausgeben |
| -i | Klein- und Großschreibung ignorieren |
| -l | nur die Namen der Datei, in denen der Text mindestens einmal gefundenen wurde, ausgeben |
| -n | Ausgabe der gefundenen Zeilen, mit vorangestellter Zeilennummer |
| -v | Alle Zeilen ausgeben, die das Suchmuster <u>nicht</u> enthalten |
| -E | Unterstützung erweiterter regulärer Ausdrücke (ERE's) (ersetzt egrep) |
| -F | Suche mit String-Konstanten (ersetzt fgrep) |
| -e muster | Die Option kennzeichnet das nachfolgende Argument als Musterbeschreibung (kann auch mit einem Minuszeichen beginnen), zur Mehrfachnennung von Suchmustern |
| -f mdatei | Suchmuster werden aus der Datei <i>mdatei</i> ausgelesen |

Es gibt drei Arten von Ausdrücken, die von **grep** unterstützt werden:

- **Fixed Regular Expressions**
String-Konstanten, wie sie von fgrep unterstützt werden. Beim grep-Kommando ist hierfür keine spezielle Option erforderlich.
- **Basic Regular Expressions (BRE)**
Einfache reguläre Ausdrücke, die standardmäßig von grep unterstützt werden.
- **Extended Regular Expressions (ERE)**
Erweiterte reguläre Ausdrücke, wie sie von egrep unterstützt werden. Beim grep-Kommando ist hierfür die Option **-E** erforderlich.

Die regulären Ausdrücke von grep (2)

- Die folgende Übersicht zeigt einen Auszug aus der Liste der regulären Ausdrücke für grep bzw. grep -E:

| grep | grep -E | Bedeutung |
|----------------|------------------|---|
| ^ | ^ | Zeilenanfang |
| \$ | \$ | Zeilenende |
| . | . | ein beliebiges Zeichen |
| [...] | [...] | eines der Zeichen aus der Liste oder aus dem Zeichenbereich (-) |
| [^...] | [^...] | ein Zeichen, das nicht in der Liste oder dem Zeichenbereich (-) steht |
| \z | \z | maskiert Metazeichen z |
| | (...) | gruppiert mehrere Zeichen |
| | | ODER-Verknüpfung |
| z* | z* | 0 bis n-malige Wiederholung von z |
| .* | .* | eine beliebige Zeichenfolge |
| | z+ | 1 bis n-malige Wiederholung von z |
| | z? | 0 oder 1-malige Wiederholung von z |

Syntax: `find startdir ... [-kriterium [arg]] ... [-aktion] ...`

Auszug aus den Kriterien:

| Kriterium | Bedeutung | | | | | | | | | | | | |
|---|--|----------|------------------|----------|--------------|----------|-----------|----------|------------------|----------|---------------|----------|------------|
| -name <i>pattern</i> | Suche nach Namen, die mit dem angegebenen Muster übereinstimmen | | | | | | | | | | | | |
| -type <i>t</i> | Suche nach Einträgen, die dem angegebenen Dateityp entsprechen <table><tr><td>f</td><td>reguläre Datei</td><td>b</td><td>Block Device</td></tr><tr><td>d</td><td>Directory</td><td>c</td><td>Character Device</td></tr><tr><td>l</td><td>symbolic Link</td><td>p</td><td>Named Pipe</td></tr></table> | f | reguläre Datei | b | Block Device | d | Directory | c | Character Device | l | symbolic Link | p | Named Pipe |
| f | reguläre Datei | b | Block Device | | | | | | | | | | |
| d | Directory | c | Character Device | | | | | | | | | | |
| l | symbolic Link | p | Named Pipe | | | | | | | | | | |
| -user <i>uname</i> | Suche nach Einträgen, die dem Nutzer <uname> gehören (uname kann UID oder Name sein) | | | | | | | | | | | | |
| -group <i>gname</i> | Suche nach Einträgen, die der Gruppe <gname> gehören (gname kann GID oder Name sein) | | | | | | | | | | | | |
| -mtime <i>[+ -]n</i> -atime <i>[+ -]n</i> -ctime <i>[+ -]n</i> | Suche nach Einträgen mit Modifikations-, Zugriffs- bzw. Statusänderungs-Zeit vor <n> Tagen (+n – n oder mehr , -n – bis zu n) | | | | | | | | | | | | |
| -newer <i>file</i> | Suche nach Einträgen, deren Modifikations- oder Statusänderungs-Zeit neuer ist als die von <file> | | | | | | | | | | | | |

- Auszug aus den Kriterien (Fortsetzung):

| Kriterium | Bedeutung |
|---|---|
| -size [+ -] <u>n</u> [c k] | Suche nach Einträgen, deren Dateigröße <n> Blöcke ist (c – Byte, k – kByte) (+n – n oder mehr, -n – bis zu n) |
| -inum <u>num</u> | Suche nach Einträgen, deren Inode-Nummer mit <num> übereinstimmt |
| -mount | durchsuche nur das Filesystem, in dem sich das <startdir> befindet |
| -perm [-] <u>onum</u> | Suche nach Einträgen, deren Zugriffsrechte genau mit <onum> übereinstimmen (-onum - es werden nur die angegebenen Rechte überprüft) |

- Mehrere Kriterien können logisch miteinander Verknüpft werden:

| Verknüpfung | Bedeutung |
|------------------|--|
| \(... \) | Gruppierung, zur Änderung des Vorrangs |
| ! | Logische Negation |
| -a | Logisches UND (Default) |
| -o | Logisches ODER |

- Der Anwender kann bestimmen, was mit den gefundenen Einträgen passieren soll:

| Aktion | Bedeutung |
|--|--|
| -print | Ausgabe der gefundenen Einträge auf stdout als Pfad; meistens die Standardaktion, wenn die Aktion in der Syntax fehlt (Ausgabeformat: <i>startdir/eintrag</i>) |
| -ls | Ausgabe der gefundenen Einträge auf stdout entsprechend dem Format des Kommandos <code>ls -l</code> |
| -exec <i>cmd</i> ... { } ... \; | Weiterverarbeitung der gefundenen Einträge mit dem <cmd> ({ } – Platzhalter für gefundenen Eintrag) |
| -ok <i>cmd</i> ... { } ... \; | Weiterverarbeitung der gefundenen Einträge mit dem <cmd> (mit interaktiver Rückfrage über stderr) |

Syntax: **sort** [*optionen*] [**-t x**] [**+pos** [**-pos**]] ... [*datei* ...]

Auszug aus den Optionen:

| Option | Bedeutung |
|----------------|---|
| -d | Sortierung nach Wörterbuchordnung (nur Buchstaben, Ziffern, Leer- und Tabulator-Zeichen sind signifikant) |
| -f | Groß- und Kleinschreibung ignorieren |
| -r | umgekehrte Sortierreihenfolge (absteigend) |
| -u | identische Zeilen nur einmal in die Ausgabe stellen |
| -o file | Ausgabe in Datei <u>file</u> statt auf Standard-Ausgabe |
| -t x | Trennzeichen für Spalten ist <u>x</u> |
| -n | numerische Sortierung |
| -b | führende Leerzeichen/Tabulatoren ignorieren |
| -M | Sortierung nach Monatskürzel „JAN“ < „FEB“ < ... < „DEC“ |

- Auszug aus den Optionen (Fortsetzung):

| Option | Bedeutung |
|---------------------------|--|
| +f1[.c] | Beginn der Sortierung hinter Feld <i>f</i> und Zeichen <i>c</i> Zeichenposition optional, Standard: Spaltenanfang |
| -f2[.c] | Ende der Sortierung hinter Feld <i>f</i> und Zeichen <i>c</i> Zeichenposition optional Standard: Spaltenanfang Endfeld optional Standard: letztes Feld |
| -k f1[.c][,f2[.c]] | Neuere Unix-/Linux-Varianten kennen diese Option, mit der die Sortierspalten exakt angegeben werden können, zum Beispiel <code>-k 4,7</code> (von Spalte 4 bis 7). Zeichenposition optional Standard: Spaltenanfang Endfeld optional Standard: letztes Feld |

Syntax: **wc [optionen] [datei(en)]**

Optionen von wc:

| Option | Bedeutung |
|-----------|---------------------------------|
| -l | (lines) Zählen von Zeilen |
| -w | (words) Zählen von Worten |
| -c | (characters) Zählen von Zeichen |

Syntax: **head [option] [datei]**

Die Option von head:

| Option | Bedeutung |
|-----------|------------------------------------|
| -n | Ausgabe der ersten <i>n</i> Zeilen |

Syntax: **tail [optionen] [datei]**

Die Optionen von tail:

| Option | Bedeutung |
|-----------|--|
| -n | Ausgabe der letzten <i>n</i> Zeilen |
| +n | Ausgabe der Datei ab Zeile <i>n</i> bis zum Ende |
| -f | Ständige Ausgabe von neu hinzugekommene Zeilen am Dateiende, bis diese mit Strg+c abgebrochen wird |

7

ZUGRIFFSRECHTE

■ Syntax: **ls -l**

```
$ ls -l /home/otto/texte
total 8
d rwx r-x r-x      otto users 512 Nov 29 12:55 briefe
d rwx r-x r-x      otto users 512 Nov 28 10:45 memo
-rw- r-- r--      otto users 267 Oct  4 09:33      vorlage1
-rw- r-- r--      otto users 861 Oct 19 07:15 briefkopf
l rwx rwx rwx      otto users  4 Nov 28 11:19 lmem -> memo
```

Gesamtgröße
in Blöcken von
allen
Dateien im Dir.

user others
└── group

Zugriffs-
rechte:
r = read
w = write
x = execute

Referenz-
zähler
(Linkcount)

Eigen-
tümer

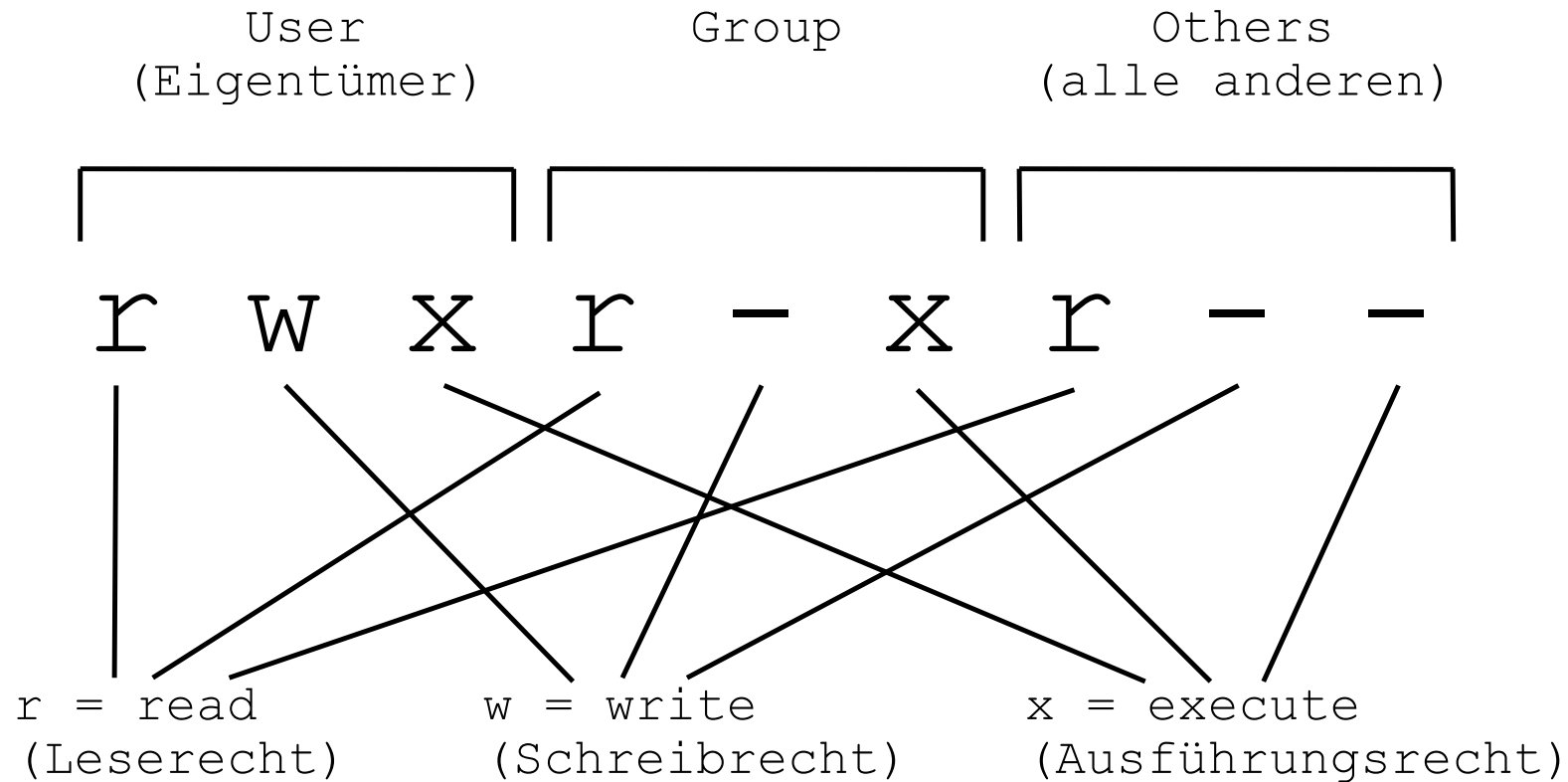
Gruppen-
zugehö-
rigkeit

Datei-
größe

Datum der
letzten
Modifikation

Datei-
name

Dateityp
(Spalte 1)



Syntax: **chmod** [*option(en)*] *rechte* *datei(en)* / *verzeichnis(se)*

Die Optionen von chmod:

| Option | Bedeutung |
|-----------|--|
| -v | (verbose) Der Befehl liefert eine ausführliche Ausgabe |
| -R | Rekursives Ändern ganzer Verzeichnisbäume; dies kann sehr gefährlich sein, da sowohl Verzeichnisse als auch Dateien gleich behandelt werden. |

Syntax: `chmod [ugo]+|-|=[rwx] datei`

➤ **Benutzerkategorien:**

- u = user (Eigentümer)
- g = group (Gruppe)
- o = other (alle anderen)
- a = all (gilt für alle 3 Klassen)

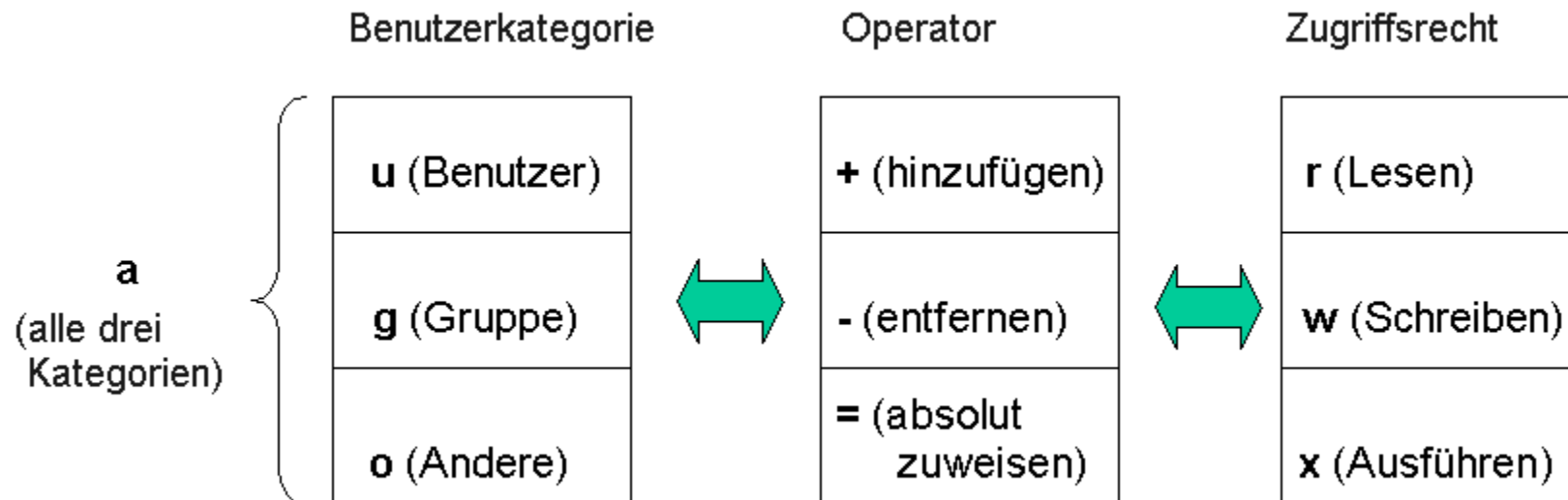
Fehlt die Angabe, gelten bei den meisten Unix- und Linux-Derivaten die Rechte für alle drei Benutzerklassen.

➤ Operator:

- + Rechte werden hinzugefügt
- - Rechte werden gelöscht
- = angegebene Rechte werden gesetzt, alle anderen gelöscht

➤ Zugriff:

- r = read
- w = write
- x = execute



Syntax: **chmod** **[n]nnn** *datei*

➤ Wertigkeit der einzelnen Rechte:

- r(ead) = 4
- w(rite) = 2
- (e)x(ecute) = 1

Jede Binärzahl lässt mit dem folgenden Schema wieder in eine Oktalzahl zurückverwandeln

| | user | group | other |
|------------|-------------|--------------|--------------|
| Rechte | r w x | r - x | - - x |
| Binärzahl | 111 | 101 | 001 |
| Wertigkeit | 4 + 2 + 1 | 4 + 0 + 1 | 0 + 0 + 1 |
| Summe | 7 | 5 | 1 |

Syntax: **umask [wert]**

| Defaultmaske des Systems | Oktalwert |
|-----------------------------|-----------|
| rw-rw-rw- für Verzeichnisse | 777 |
| rw-rw-rw- für Dateien | 666 |

nach Befehl ...

| | | |
|------------------|-----------------------------|-----|
| umask 027 | rw-r-x--- für Verzeichnisse | 750 |
| | rw-r----- für Dateien | 640 |
| umask 022 | rw-r-xr-x für Verzeichnisse | 755 |
| | rw-r--r-- für Dateien | 644 |
| umask 077 | rw-x----- für Verzeichnisse | 700 |
| | rw----- für Dateien | 600 |
| umask 002 | rw-rwxr-x für Verzeichnisse | 775 |
| | rw-rw-r-- für Dateien | 664 |

- ❑ **Gruppenzugehörigkeit ermitteln** mit dem Befehl **groups**:

groups [*benutzer*]

- ❑ **Benutzername, UID, Gruppen und GIDs ermitteln** mit dem Befehl **id**:

id [-a] [*benutzer*]

- ❑ **Gruppe der Datei ändern** mit dem Befehl **chgrp**:

chgrp [-R] *gruppe datei(en)*

- ❑ **Das Setzen des SGID-Bit** erfolgt mit dem **chmod**-Kommando:

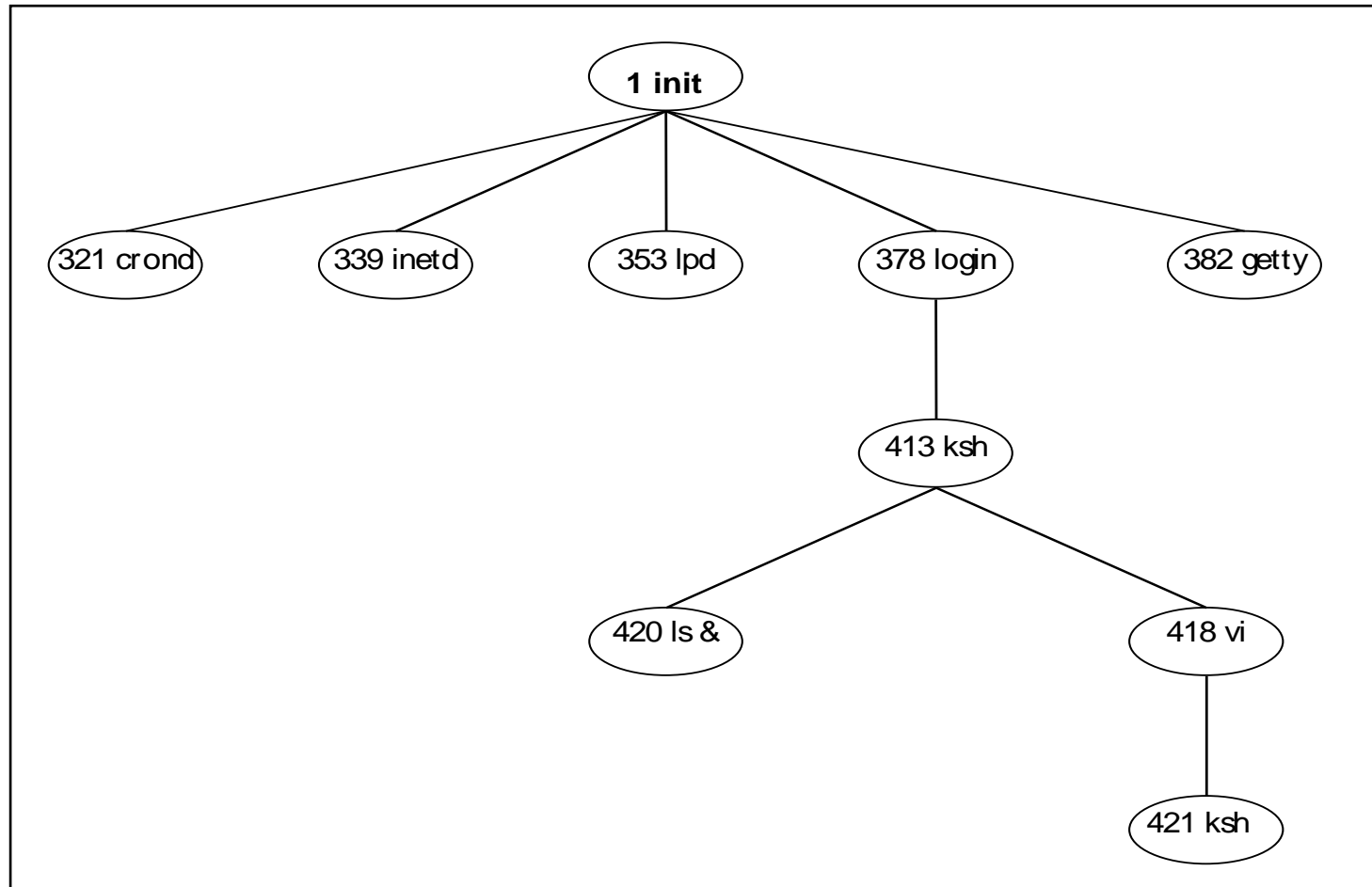
Symbolische Methode: **g+s**

Oktale Methode: **2** (an der Tausender-

Position)

8

PROZESSVERWALTUNG



- Syntax: **ps** *[optionen]*

Die wichtigsten Optionen von ps:

| Option | Bedeutung |
|---------------------------|---|
| -e | (extended) Alle Prozesse im System werden angezeigt |
| -f | (full) Ausführliche Ausgabe zum Prozess |
| -l | (long) Ausgabe weiterer Informationen |
| -u <i>user</i> | (user) Alle Prozesse des angegebenen Benutzers |
| -t <i>terminal</i> | (terminal) Alle Prozesse des angegebenen Terminals |
| -p <i>pid</i> | (pid) Nur Prozess mit angegebener PID auswählen |

Signale sind “Mitteilungen” an einen Prozess

Zu den wichtigsten Signalen gehören:

| Nummer | Signal | Beschreibung |
|--------|-------------|---|
| 1 | HUP | Hangup (Ausschalten eines direkt angeschlossenen Terminals, Beenden der Anmeldeshell) |
| 2 | INT | Interrupt (in der Regel ausgelöst durch Ctrl-c) |
| 9 | KILL | Kill (das einzige nicht abfangbare Signal, führt zu sofortigem Prozessabbruch) |
| 15 | TERM | Software-Beendigung (Standardsignal) |

➤ **kill** [-nummer] *pid(s)*

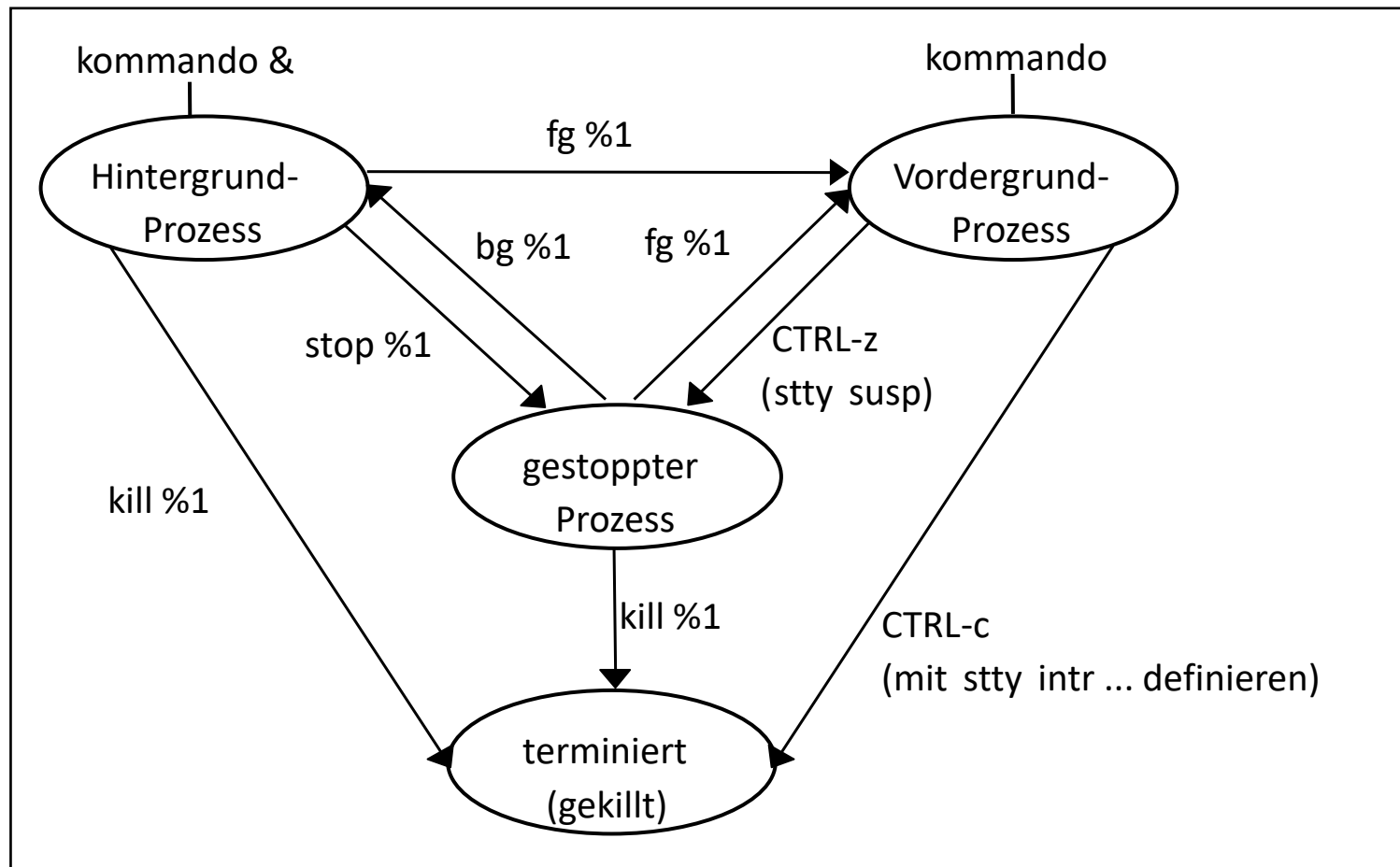
➤ **kill** [-NAME] *pid(s)*

➤ **kill** *pid*

“Harter“ Prozessabbruch:

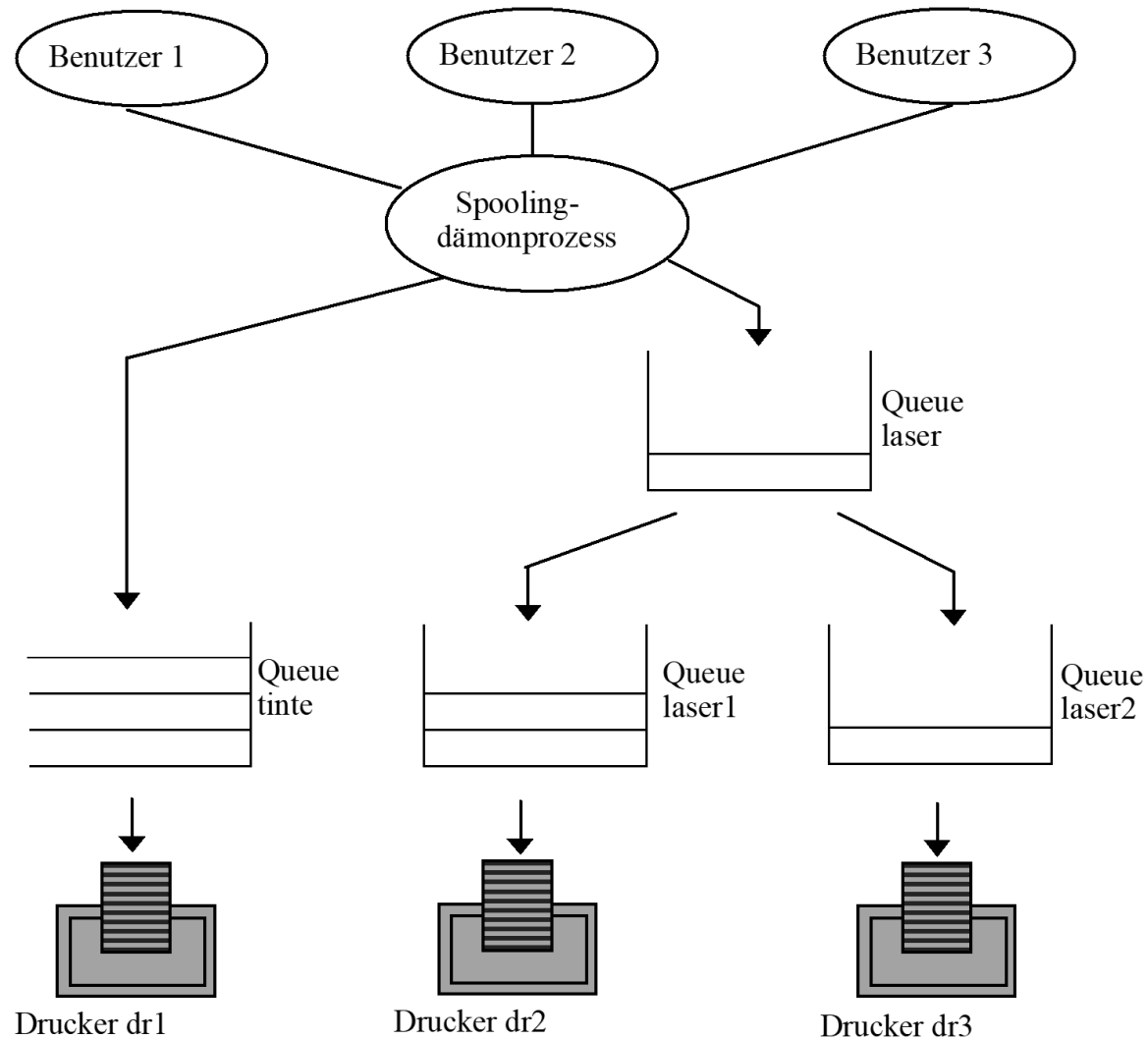
➤ **kill -9** *pid*

- Hintergrundprozesse anzeigen mit dem Befehl jobs



9

DRUCKEN



Vorteile:

- Kann Benutzer über Passwörter oder Zertifikate authentifizieren
- Kann die zu druckenden Daten bei der Übertragung zwischen zwei Rechnern oder zum Drucker verschlüsseln
- Kann alle im Netz zur Verfügung stehenden Drucker jedem Client-Rechner beim Booten oder Anmelden mitteilen

Die Befehle **lpr** und **lp** sind weiterhin gültig

CUPS steht zur Verfügung für:

- alle Linux-Versionen ab Kernelversion 2.0 (Intel-Prozessoren)
- Solaris (für die Intel- und SPARC-Versionen) von Sun Microsystems ab Version 2.5 bis zur Version 9
- HP-UX von Hewlett-Packard ab Version 10.20 bis zur Version 11.0
- IRIX von Silicon Graphics Inc. ab Version 5.3 bis zur Version 6.5
- Digital Unix von Digital Equipment Corporation (heute Compaq) ab Version 4.0
- Tru64 Unix von Compaq ab Version 4.0
- OSF/1 ab Version 4.0.

➤ **Dokumente ausdrucken – der Befehl lpr**

lpr [**-P** *printer*] [*optionen*] [*datei1* [*datei2* ...]]

➤ **Einsehen der Druckerwarteschlange – der Befehl lpq**

lpq [**-l**] [**-P** *printer*] [*user*]

➤ **Druckaufträge löschen – der Befehl lprm**

lprm [**-P** *printer*] [**-**] [*jobid*, ...]

➤ **Dokumente ausdrucken – der Befehl lp**

lp [**-d** *printer*] [*optionen*] [*datei1* [*datei2 ...*]]

➤ **Einsehen der Druckerwarteschlange – der Befehl lpstat**

lpstat [*printer*] [*optionen*]

➤ **Druckaufträge löschen – der Befehl cancel**

cancel [*printer*] [*jobid ...*]

10

RECHNERÜBERGREIFENDES ARBEITEN

➤ **Rechnername ausgeben**

hostname

uname -n

➤ **Testen der Verbindung**

ping *rechnername* | *ip-adresse*

Befehle für das Arbeiten und Kopieren in einem heterogenen Netzwerk mit verschiedenen Betriebssystemen:

- **Remote Anmelden und Arbeiten** mit dem Befehl **telnet**

telnet *hostname* / *ip-adresse*

!!! telnet ist nicht verschlüsselt !!!

- **Remote Kopieren** mit dem Befehl **ftp**

ftp *hostname* | *ip-adresse*

Vorteile:

- Verschlüsselte Übertragung der Passworte
- Sicherung von Authentizität und Integrität der übertragenen Daten

- Remote Anmelden bzw. Remote Kommando ausführen - ssh

ssh *hostname* | *ip-adresse* [-l *user*] [*kommando*]

ssh *hostname@user* [*kommando*]

- Remote Kopieren - scp

scp [-r] [*user@*][*hostname:*][*path*] [*user@*][*hostname:*][*path*]

11

SONSTIGE BEFEHLE

- Der Befehl **printf** format [arg/ arg2 ...]

Escape-Sequenzen im Formatierungsstring

| Escape-Sequenz | Bedeutung |
|----------------|-------------------------|
| \n | Newline (Zeilenumbruch) |
| \r | Return |
| \t | Tabulator |
| \v | Vertikaler Tabulator |
| \a | Alarm (System-Piep) |
| \\ | Backslash |

- Der Befehl **printf** format [arg/ arg2 ...]

Platzhalter im Formatierungsstring

| Platzhalter | Bedeutung |
|-------------|---|
| Text | Unveränderte Ausgabe von „Text“ |
| %s | Platzhalter für einen String (bzw. eine Zahl) |
| %ms | Platzhalter für einen <i>rechtsbündig</i> ausgegeben String mit einer Breite von <i>m</i> Zeichen |
| %-ms | Platzhalter für einen <i>linksbündig</i> ausgegeben String mit einer Breite von <i>m</i> Zeichen |
| %m.ns | Platzhalter für einen String mit einer Gesamtbreite von <i>m</i> Zeichen. Vom String selbst werden jedoch nur <i>n</i> Zeichen ausgegeben |
| %m.nf | Platzhalter für einen Gleitkommazahl mit einer Gesamtbreite von <i>m</i> Zeichen und <i>n</i> Nachkommastellen |

Anzapfen einer Pipe mit dem Befehl tee:

- Bewirkt ein zusätzliches Schreiben auf die Standardausgabe
- Beispiel: `ls | tee liste | wc -l`

Spalten ausschneiden mit dem Befehl cut:

Syntax: **cut** [*optionen*] *datenstrom*

Die Optionen von **cut**:

| Option | Bedeutung |
|------------|--|
| -f | (field) Die angegebenen Felder werden ausgewählt, wobei der Tabulator das Feldtrennzeichen ist |
| -c | (columns) Die angegebenen Spalten werden ausgewählt |
| -dx | (delimiter) Das Feldtrennzeichen ist x |

Zeichenumwandlung mit dem Befehl tr:

Syntax: **tr** [*optionen*] *zfolge1* [*zfolge2*]

Die Optionen von **tr**:

| Option | Bedeutung |
|-----------|--|
| -d | (delete) Die angegebenen Zeichen aus <i>zfolge1</i> werden gelöscht |
| -s | (squeeze) In der ersten Folge mehrfach hintereinander auftretende Zeichen werden nur einmal abgebildet |
| -c | (complement) alle Zeichen außer denen in der ersten Folge werden umgewandelt |

Formatierte Ausgabe für den Druck mit dem Befehl **pr**:

Syntax: **pr** *[optionen]* *[datei]*

Die Optionen von **pr**:

| Option | Bedeutung |
|-------------------|---|
| -n | Ausgabe erfolgt <i>n</i> -spaltig |
| +n | Ausdruck beginnt auf Seite <i>n</i> |
| -wn | (width) Zeilenbreite ist <i>n</i> Zeichen (Default: 72) |
| -on | linken Rand um <i>n</i> Zeichen einrücken |
| -ln | (length) Seitenlänge ist <i>n</i> Zeilen (Default: 66) |
| -h "titel" | (header) Als Titel wird nicht der Dateiname, sondern <i>titel</i> verwendet |
| -t | Unterdrückung von Kopf- und Fußzeilen, keine Leerzeilen am Seitenende, kein Form Feed |

- **Festplattenbelegung überprüfen** mit dem Befehl **df** *[optionen]*

Optionen von df

| Option | Bedeutung |
|---------|---|
| (keine) | Ausgabe aller verfügbaren Dateisysteme |
| -k | Ausgabe in Kilobytes (auf einigen Derivaten wie z. B. Linux Default) |
| -h | Human readable: Ausgabe in besser lesbarer Form (Giga-/Mega-/Kilobyte) |

- **Platzbedarf berechnen** mit dem Befehl **du** *[option]* *[verzeichnis]*

Optionen von du

| Option | Bedeutung |
|--------|--|
| -a | all: auch alle Dateien werden ausgegeben |
| -s | nur die Summe wird ausgegeben |



at *zeit [datum] [+ inkrement]*
kommando(s)
CTRL d

Zeitangabemöglichkeiten

Beispiele zur Uhrzeit

| | |
|--------------------|-------------------------|
| at 11 < datei | Ausführung um 11 Uhr |
| at 9:32 < datei | Ausführung um 09:32 Uhr |
| at 9pm < datei | Ausführung um 21 Uhr |
| at noon < datei | Ausführung um 12 Uhr |
| at teatime < datei | Ausführung um 16 Uhr |

Beispiele zu Uhrzeit und Datum

| | |
|-------------------------|--|
| at 11am Thu < datei | Ausführung am nächsten Donnerstag 11 Uhr |
| at 7:30 Sep 24 < datei | Ausführung am 24. September um 07:30 |
| at 12:45 Sunday < datei | Ausführung am Sonntag um 12:45 Uhr |

Beispiele zu Uhrzeit, Datum und Inkrement

| | |
|---------------------------------|--|
| at now + 3 days < datei | Ausführung in genau 3 Tagen |
| at 15:45 Fri + 9 months < datei | Ausführung am Freitag 15:45 in 9 Monaten |

■ crontab-Dateiformat

| Minuten | Stunden | Tag | Monat | Wochentag | Kommando |
|---------|---------|-----|-------|-----------|----------|
|---------|---------|-----|-------|-----------|----------|

Die ersten 5 Felder definieren Datum und Uhrzeit der Kommandoausführung:

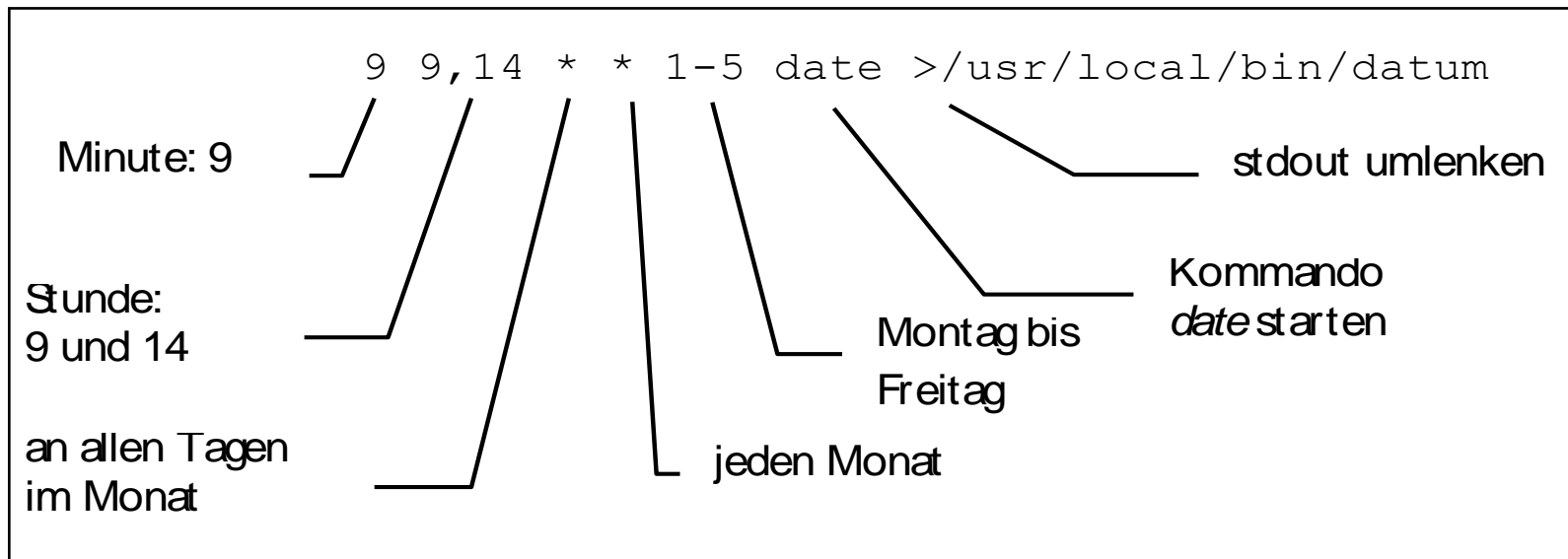
1. Minuten (0-59)
2. Stunden (0-23)
3. Tag im Monat (1-31)
4. Monat (1-12)
5. Wochentag (0-7, 0 = 7 = Sonntag)

Jedes Feld kann enthalten:

- eine Zahl im eben spezifizierten Bereich
- zwei Zahlen durch ein Minus getrennt (bedeutet einschließlich)
- Zahlen durch Kommata getrennt (bedeutet alle angegebenen Werte)
- ein Stern (bedeutet alle möglichen Werte)
- Das sechste Feld schließlich ist ein Kommando, welches zu den in den ersten 5 Feldern festgelegten Zeiten ausgeführt wird.

■ crontab-Dateiformat

Beispiel:



Editieren von Crontabs

- **crontab** *[option]* *[user]*

Optionen von crontab

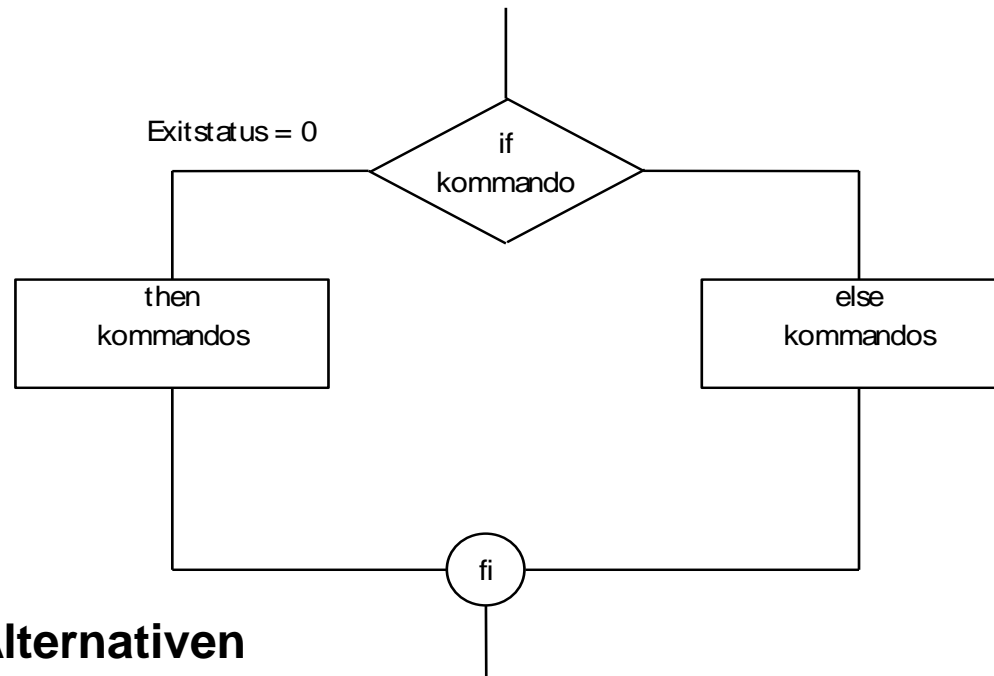
| Option | Bedeutung |
|--------|--|
| -l | Ausgabe der aktuellen crontab auf den Bildschirm |
| -e | Editieren der crontab mit dem über \$EDITOR eingestellten Editor |
| -r | Löschen der crontab |

12

EINFÜHRUNG IN SHELLPROZEDUREN

| Aufruf | nötige Rechte | Beschreibung |
|---------------------|----------------|--|
| ksh prozedur | read | Start einer Subshell (hier Korn-Shell), welche die Befehle aus der Prozedur liest |
| prozedur | read + execute | Start der Prozedur mit ihrem Namen Voraussetzung: x-Recht ist gesetzt |
| . prozedur | read | Der Punktbefehl: Die aktuelle Shell öffnet die Prozedur und führt die Kommandos aus |

| | |
|---------------|--|
| \$0 | Name der Kommandoprozedur |
| \$1 | 1. Positionsparameter |
| \$2 | 2. Positionsparameter |
| \$9 | 9. Positionsparameter |
| \${10} | 10. Positionsparameter |
| \${n} | <i>n</i> . Positionsparameter |
| \$* | Liste aller Positionsparameter (\$1...\$n) |
| \$# | Anzahl der übergebenen Parameter |
| \$\$ | Prozess-ID des aktuellen Prozesses |
| \$? | Exitstatus des letzten Kommandos |



Syntax-Alternativen

if kommando
then kommando(s)
fi

if *kommando*
then *kommando(s)*
else *kommando(s)*
fi

Stringvergleiche

| | |
|---|---|
| [[<i>string</i>]] | wahr, wenn kein Nullstring |
| [[-n <i>string</i>]] | dito |
| [[-z <i>string</i>]] | wahr, wenn String ein Nullstring ist (leer) |
| [[<i>str1</i> = <i>str2</i>]] | wahr, wenn beide Strings identisch sind |
| [[<i>str1</i> != <i>str2</i>]] | wahr, wenn beide Strings ungleich sind |

Vergleich von Dateiattributen

| | |
|------------------------------|---|
| [[-f <i>datei</i>]] | wahr, wenn die Datei eine normale Datei ist |
| [[-d <i>datei</i>]] | wahr, wenn Directory |
| [[-L <i>datei</i>]] | wahr, wenn symbolischer Link |
| [[-s <i>datei</i>]] | wahr, wenn Größe > 0 |
| [[-r <i>datei</i>]] | wahr, wenn Datei vom Prozess lesbar ist |
| [[-w <i>datei</i>]] | wahr, wenn Datei vom Prozess beschreibbar |
| [[-x <i>datei</i>]] | wahr, wenn Datei vom Prozess ausführbar |

Numerische Vergleiche (Integer)

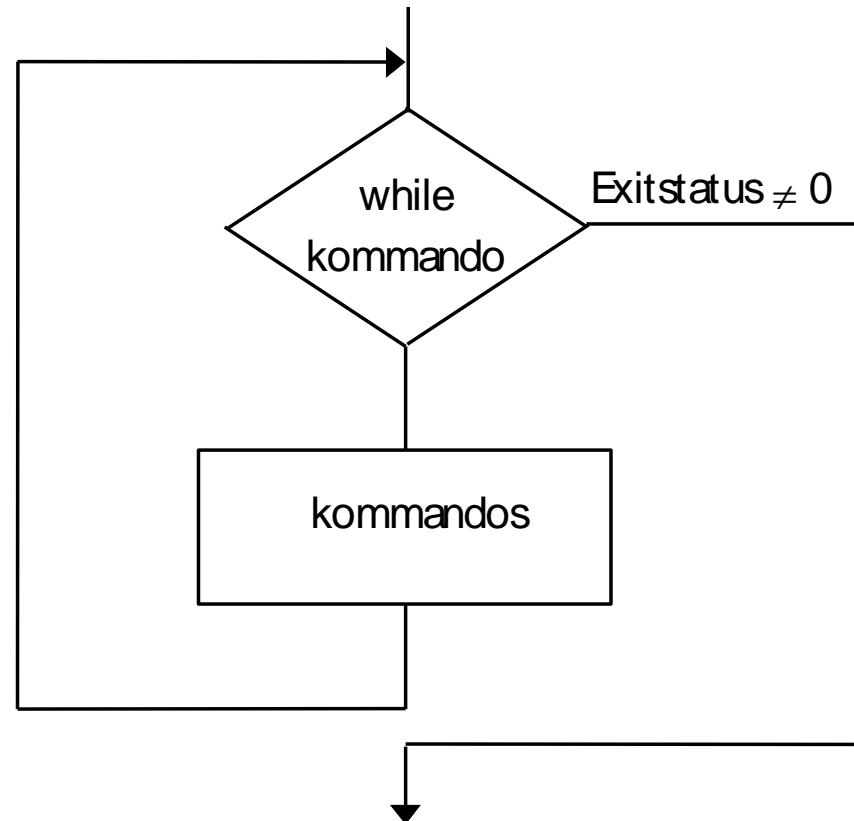
| | |
|------------------------------|--|
| [[wert1 -eq wert2]] | wahr, wenn beide Integerwerte gleich |
| [[wert1 -ne wert2]] | wahr, wenn beide Integerwerte ungleich |
| [[wert1 -gt wert2]] | wahr, wenn wert1 größer als wert2 |
| [[wert1 -ge wert2]] | wahr, wenn wert1 größer gleich wert2 |
| [[wert1 -lt wert2]] | wahr, wenn wert1 kleiner als wert2 |
| [[wert1 -le wert2]] | wahr, wenn wert1 kleiner gleich wert2 |

Verknüpfung mehrerer Bedingungen

| | |
|---------------------------|--|
| \(... \) | Klammerung von Ausdrücken, um Vorrangregeln zu verändern |
| [[! bed1]] | Negation einer Bedingung |
| [[bed1 -a bed2]] | UND-Verknüpfung zweier Bedingungen |
| [[bed1 -o bed2]] | ODER-Verknüpfung zweier Bedingungen |

Syntax:

```
while befehl  
do      befehl(e)  
done
```



Syntax:

for *variable in element1 element2 ...*

do

befehl(e)

done

