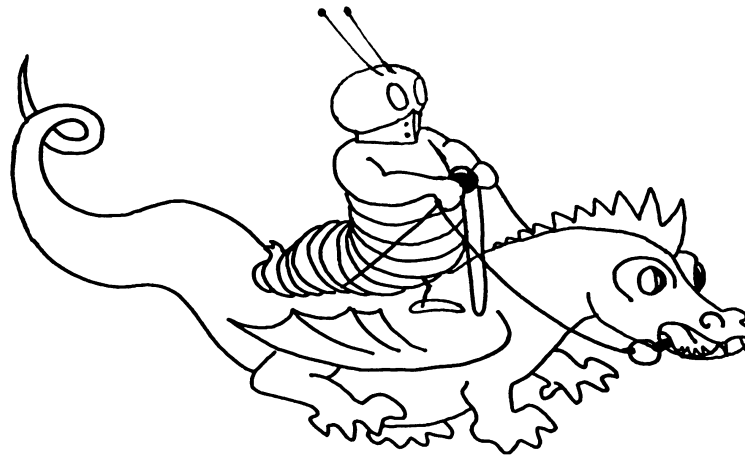


# Linux/Unix

Grundlagen der Administration  
von Christine Wolfinger  
in Zusammenarbeit mit



*Viel Freude beim Lernen*

# Übersicht UNIX/Linux Einführung für Administratoren

| Themen                             | Um was geht's dabei?   |
|------------------------------------|--|
| 1 Linux/Unix Grundkonzepte         | Einordnung der Betriebssysteme<br>Besonderheiten von UNIX/Linux                          |
| 2 Einführung in Linux/Unix         | Auf los geht's los<br>Anmelden, Benutzer unter UNIX,<br>Eingabe von Kommandos (Terminal) |
| 3 Die grafische Benutzeroberfläche | Wie arbeiten Sie mit der grafischen<br>Umgebung unter Linux                              |
| 4 Dateiverwaltung                  | Dateien, Directories (Ordner, Verzeichnisse), Datei-<br>baum, Kommandos zur Verwaltung   |
| 5 Zugriffsberechtigungen           | Wer darf was (lesen, schreiben, ausführen)   |
| 6 Der Editor vi / vim und sed      | Ein alter - aber wichtiger Editor  |
| 7 Bash Eingabe/Ausgabeumlenkung    | Grundlagen der Bash. Wohin mit dem Ergebnis<br>(Terminal, Datei, Weiterverwendung)       |
| 8 Bash - Metazeichen/Suchmuster    | Schreibarbeit sparen, Ersetzungs- und<br>Filtermechanismen                               |

| Themen |                                      | Um was geht's dabei?   |
|--------|--------------------------------------|--|
| 9      | Erleichterungen mit der Bash         | Noch einfacheres und sicheres Arbeiten                                     |
| 10     | Variablen- und Kommandosubstitution  | Raffinierte Verknüpfungen, Ersetzungsmechanismen                           |
| 11     | Werkzeuge/Dienstprogramme und Filter | Schnelle Ergebnisse durch Kombination von Befehlen                         |
| 12     | Shell-Prozeduren                     | Sich wiederholende Kommandos in einem kleinem Programm zusammenfassen      |
| 13     | Überprüfungen von Bedingungen        | Abfragen auf: Erfolgreich - Nicht erfolgreich, Vorhanden - Nicht vorhanden |
| 14     | Ablaufsteuerung                      | „Wenn das so ist, tue ...“<br>„Tue solange bis ...“                        |
| 15     | Drucken unter Linux                  | Das CUPS-System und Druckkommandos   |
| 16     | Systemüberwachung                    | Welche Prozesse sind z.Z. aktiv. Welche Informationen können Sie abrufen.  |
| 17     | Hilfesysteme unter Linux/Unix        | Wo und wie finden Sie Hilfe – Hilfe zur Selbsthilfe                        |

# UNIX/Linux Einführung für Administratoren

|   |              |  |              |
|---|--------------|--|--------------|
| Übersicht UNIX/Linux Einführung für Administratoren.....        | 1 - II       | am schnellsten geht's übers Terminal.....              | 3 - 6        |
| <b>1 Linux/Unix Grundkonzepte .....</b>                         | <b>1 - 1</b> | Dateiverwaltung mit dem Konqueror .....                | 3 - 7        |
| Betriebssystemarten .....                                       | 1 - 2        | Nur die wichtigsten Funktionen .....                   | 3 - 8        |
| Was ist das Besondere an UNIX/LINUX? .....                      | 1 - 3        | Einige Begriffe und Symbole.....                       | 3 - 9        |
| Multi-User und Multi-Tasking .....                              | 1 - 4        | Suchfunktionen .....                                   | 3 - 10       |
| Aufgaben des UNIX Betriebssystems.....                          | 1 - 5        | Bereits vorhandene Dateien im Home-Directory .....     | 3 - 11       |
| Einordnen der Shell (Bash).....                                 | 1 - 6        | Wie legen Sie ein neues Verzeichnis an? .....          | 3 - 12       |
| Welche Portierungen gibt es von UNIX .....                      | 1 - 7        | Verschieben, Kopieren und Löschen von Dateien.....     | 3 - 13       |
| Einsatzmöglichkeiten von Linux.....                             | 1 - 8        | Abmelden .....   | 3 - 14       |
| Unterschiede von Linux zu Windows.....                          | 1 - 10       | <b>4 Das Dateisystem von UNIX .....</b>                | <b>4 - 1</b> |
| <b>2 Auf los geht's los .....</b>                               | <b>2 - 1</b> | Welche Dateinamen können Sie vergeben?.....            | 4 - 2        |
| Wie melden Sie sich an.....                                     | 2 - 2        | Absolute und relative Pfadangabe .....                 | 4 - 3        |
| Root und „normale Benutzer“ .....                               | 2 - 3        | Wie sehen Sie, wo Sie sind? .....                      | 4 - 4        |
| Beispiel eines Teilausschnittes der /etc/passwd.....            | 2 - 4        | Dateien und Directories .....                          | 4 - 5        |
| Was bedeuten die Zeichen auf dem Bildschirm?.....               | 2 - 5        | Anzeige von Dateien.....                               | 4 - 6        |
| Abmelden .....  | 2 - 6        | Klettern durch den Dateibaum.....                      | 4 - 7        |
| Wie schreiben Sie Kommandos ? .....                             | 2 - 7        | Kommandos für die Tour.....                            | 4 - 8        |
| Syntax von UNIX Kommandos.....                                  | 2 - 8        | Erstellen von Directories? .....                       | 4 - 11       |
| Einige Kommandos.....   | 2 - 9        | Löschen von Directories.....                           | 4 - 11       |
| Verkettung von Kommandos ";" .....                              | 2 - 10       | Weitere Kommandos zur Dateiverwaltung .....            | 4 - 12       |
| Zusammenfassung .....   | 2 - 11       | Suchen von Zeilen innerhalb einer Datei .....          | 4 - 17       |
| <b>3 Ein kleiner Blick in die grafische Oberfläche KDE3 - 1</b> |              | Formatierte Ausgabe .....                              | 4 - 18       |
| Wie melden Sie sich an.....                                     | 3 - 2        | Ansehen von Dateiinhalten (less, more oder pg) .....   | 4 - 19       |
| Der Desktop - Ihr Arbeitsbereich.....                           | 3 - 3        | Nur ein kleiner Blick in die Dateien: .....            | 4 - 20       |
| Wie starten Sie Programme - wo ist was .....                    | 3 - 4        | find – Kommando, um Dateien zu suchen und zu finden .. | 4 - 21       |
| Ob auf dem Desktop oder unter Startmenü .....                   | 3 - 5        | find mit grep kombiniert.....                          | 4 - 23       |
|   |              | <b>5 Zugriffsrechte .....</b>                          | <b>5 - 1</b> |
|   |              | Verändern von Zugriffsrechten .....                    |              |

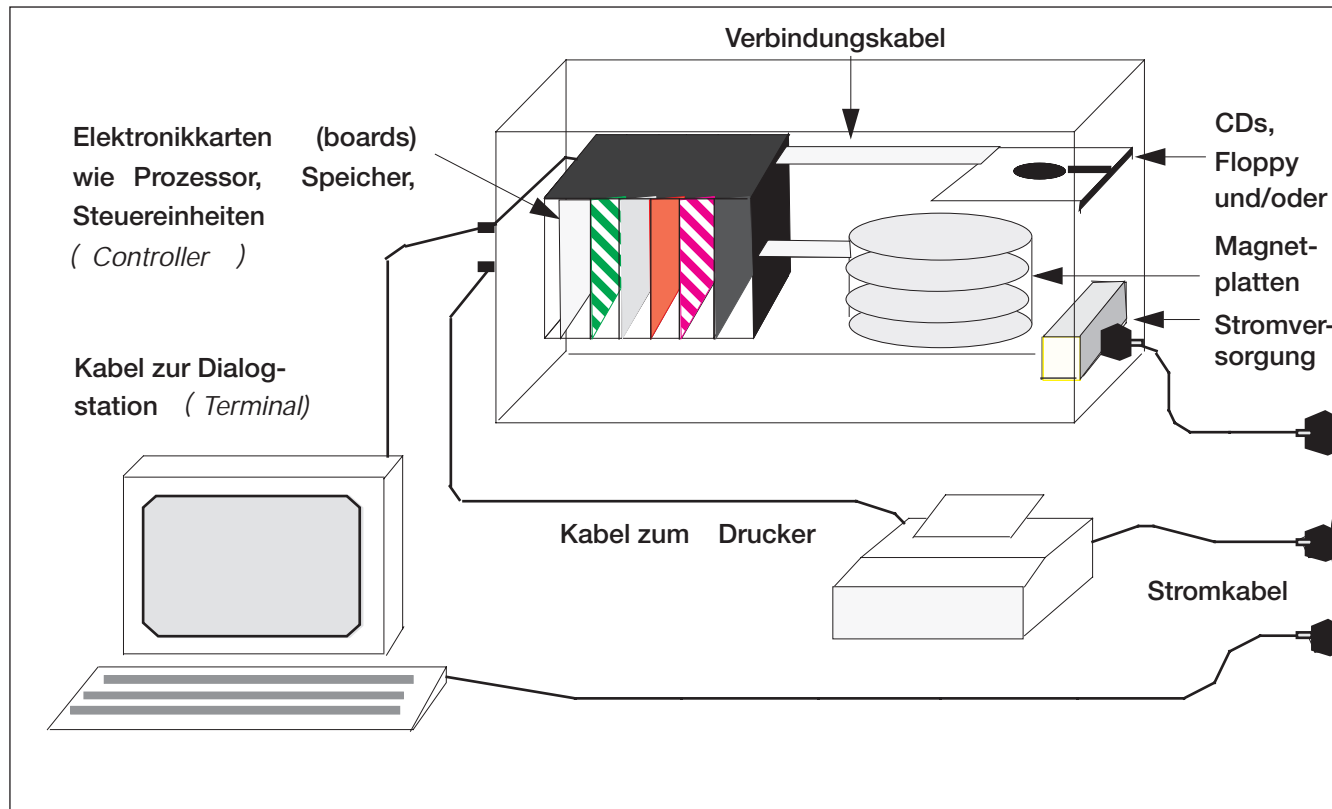
|  |              |   |               |
|--|--------------|---|---------------|
| mit Symbolen.....                                  | 5 - 2        | Anhängen an eine Datei .....                            | 7 - 4         |
| Ändern von Zugriffsrechten                         |              | Eingabeumlenkung.....                                   | 7 - 5         |
| mit Oktalzahlen.....                               | 5 - 3        | Fehlerumlenkung.....                                    | 7 - 6         |
| Besitzer- und Gruppenwechsel.....                  | 5 - 4        | Pipelining .....  | 7 - 7         |
| Sonderrechte SUID/SGID-Bit.....                    | 5 - 5        | Filterkommandos .....                                   | 7 - 8         |
| Sonderrecht Sticky-Bit (t).....                    | 5 - 6        | sort – Kommando, um Dateiinhalte zu sortieren .....     | 7 - 8         |
| Oktalzuordnung von SUID/SGID und Sticky-Bit .....  | 5 - 7        | grep – Kommando .....                                   | 7 - 9         |
| Voreinstellung über umask.....                     | 5 - 8        |   |               |
| <b>6 Der Editor vi und sed.....</b>                | <b>6 - 1</b> | <b>8 Shell - Metazeichen/Suchmuster .....</b>           | <b>8 - 1</b>  |
| Wie arbeiten Sie mit dem vi?.....                  | 6 - 2        | Aufhebung eines Sonderzeichens (z.B. Metazeichen) ..... | 8 - 2         |
| Wie arbeiten Sie mit dem vi?.....                  | 6 - 3        | Weitere Ersetzungsmechanismen.....                      | 8 - 3         |
| Wie arbeiten Sie mit dem vi?.....                  | 6 - 4        |   |               |
| Wie positionieren Sie den Cursor? .....            | 6 - 5        | <b>9 Erleichterungen mit der Bash .....</b>             | <b>9 - 1</b>  |
| Wie blättern Sie in der Datei?.....                | 6 - 6        | Übersicht einiger Shells .....                          | 9 - 2         |
| Wie geben Sie Text ein? .....                      | 6 - 7        | Vorteile der bash (bzw. ksh).....                       | 9 - 3         |
| Korrekturen im Eingabemodus.....                   | 6 - 8        | Starten und Abmelden einer bash (bzw ksh).....          | 9 - 4         |
| Wie korrigieren Sie im Befehlsmodus .....          | 6 - 9        | Der Historie-Mechanismus .....                          | 9 - 5         |
| Wie löschen Sie im Befehlsmodus? .....             | 6 - 10       | Unter Bash einfache Korrekturen und Ergänzungen .....   | 9 - 6         |
| Wie kopieren und fügen Sie ein?.....               | 6 - 11       | Kommandozeilen in der ksh editieren? .....              | 9 - 7         |
| Wie suchen Sie nach Begriffen?.....                | 6 - 12       | Weitere vi-Funktionen.....                              | 9 - 8         |
| Einige im vi verwendete Kürzel.....                | 6 - 13       | Expandierung von Pfadnamen.....                         | 9 - 9         |
| Sonstige Hilfen.....                               | 6 - 14       | Tilde-Expandierung (für bash und ksh).....              | 9 - 10        |
| Einige praktische Optionen für den vi .....        | 6 - 15       | Die Alias-Funktion .....                                | 9 - 11        |
| SED.....   | 6 - 16       | Alias-Befehle .....                                     | 9 - 12        |
| Funktionen innerhalb des Scripts .....             | 6 - 17       | Zusätzliche Optionen.....                               | 9 - 13        |
| Reguläre Ausdrücke - Metazeichen/Wildcards .....   | 6 - 18       | Verwendung von Optionen .....                           | 9 - 14        |
| <b>7 Bash Shell Eingabe/Ausgabeumlenkung .....</b> | <b>7 - 1</b> | <b>10 Variablen- und Kommandosubstitution.....</b>      | <b>10 - 1</b> |
| Ausgabeumleitung .....                             | 7 - 2        | Was sind Variable?.....                                 | 10 - 1        |
|  |              | Wie werden Variable zugeordnet? .....                   | 10 - 2        |

|   |               |   |               |
|---|---------------|---|---------------|
| Wie arbeiten Sie mit Variablen? .....                 | 10 - 3        | Eigene Kommandos .....                        | 12 - 4        |
| Einige vordefinierte Variable .....                   | 10 - 7        | Ergänzen des Suchpfades .....                 | 12 - 6        |
| Wie kann das Promptzeichen verändert werden? .....    | 10 - 9        | Wie verwenden Sie Positionsvariable? .....    | 12 - 7        |
| Löschen von gesetzten Variablen.....                  | 10 - 9        | Welche Positionsparameter gibt es? .....      | 12 - 8        |
| Kommandosubstitution.....                             | 10 - 10       | Dialog mit dem Benutzer .....                 | 12 - 9        |
| Ersetzungsmechanismen.....                            | 10 - 12       | <b>13 Überprüfungen von Bedingungen .....</b> | <b>13 - 1</b> |
| Voreinstellung für Bash und Korn-Shell .....          | 10 - 13       | Das test-Kommando .....                       | 13 - 1        |
| Der Datentyp integer.....                             | 10 - 14       | Zeichenketten- und Wertetest .....            | 13 - 4        |
| <b>11 Werkzeuge und Dienstprogramme .....</b>         | <b>11 - 1</b> | test - moderne Schreibweise.....              | 13 - 6        |
| Suchen von Zeilen innerhalb einer Datei.....          | 11 - 1        | Kombination von test .....                    | 13 - 6        |
| Metazeichen – reguläre Ausdrücke .....                | 11 - 2        | Algebraische Vergleiche .....                 | 13 - 7        |
| find – Kommando, um Dateien zu suchen und zu finden.. | 11 - 5        | <b>14 Ablaufsteuerung .....</b>               | <b>14 - 1</b> |
| find mit grep kombiniert .....                        | 11 - 7        | Die if-Abfrage .....                          | 14 - 2        |
| Ausschneiden von Spalten – cut .....                  | 11 - 8        | Die case-Anweisung.....                       | 14 - 2        |
| Das Sort-Programm .....                               | 11 - 10       | Die for-Schleife .....                        | 14 - 4        |
| Formatierte Ausgabe .....                             | 11 - 12       | Die while- und until-Schleife .....           | 14 - 6        |
| Ansehen von Dateiinhalten.....                        | 11 - 13       | Beispiel einer kleinen Shell-Prozedur .....   | 14 - 8        |
| Nur ein kleiner Blick in die Dateien: .....           | 11 - 15       | Vorzeitiges Beenden von Shellprozeduren.....  | 14 - 9        |
| Datenstrom mit tee duplizieren.....                   | 11 - 16       | Einige Shell-interne Kommandos.....           | 14 - 10       |
| Dateien vergleichen .....                             | 11 - 17       | <b>15 Drucken unter Linux .....</b>           | <b>15 - 1</b> |
| Dateien drucken .....                                 | 11 - 18       | Verwalten von Druckaufträgen.....             | 15 - 2        |
| Kontrolle über Druckaufträge .....                    | 11 - 19       | Topologie des CUPS-Systems .....              | 15 - 3        |
| Abbruch von Druckaufträgen .....                      | 11 - 19       | Druckkommandos lpr .....                      | 15 - 4        |
| Alternative Druckkommandos .....                      | 11 - 20       | Kontrolle mit lpc .....                       | 15 - 5        |
| <b>12 Shell-Prozeduren .....</b>                      | <b>12 - 1</b> | Druckkommandos lp .....                       | 15 - 6        |
| Was ist ein Shell-Skript (Shell-Prozedur)?.....       | 12 - 1        | <b>16 Systemüberwachung .....</b>             | <b>16 - 1</b> |
| Wie erstellen Sie eigene Kommandos? .....             | 12 - 3        | Jobcontrol - Prozeßverwaltung.....            | 16 - 1        |

|  |               |  |        |
|--|---------------|--|--------|
| Wie sehen Sie, welche Prozesse vorhanden sind? .....     | 16 - 2        | Online-Foren, Usergroups und Newsgroups..... | 17 - 7 |
| Wie können Sie bestimmte Prozesse auswählen?.....        | 16 - 4        | Notfall-CD und Knoppix .....                 | 17 - 8 |
| Prozeßattribute .....                                    | 16 - 5        | Wichtige Tastaturkombinationen .....         | 17 - 9 |
| Weitere Kommandos zur Anzeige von Prozessen .....        | 16 - 7        |  |        |
| Prioritäten verändern.....                               | 16 - 8        |  |        |
| Beispiel einer Jobsteuerung .....                        | 16 - 10       |  |        |
| Kommandos zur Prozeßsteuerung.....                       | 16 - 11       |  |        |
| Wie können Sie mehrere Prozesse killen? .....            | 16 - 13       |  |        |
| Prozesskontrolle unter der KDE .....                     | 16 - 14       |  |        |
| At-Kommandos .....                                       | 16 - 16       |  |        |
| Hinweise zu at .....                                     | 16 - 17       |  |        |
| Zeitangaben mit at.....                                  | 16 - 18       |  |        |
| Zeitgesteuerte Arbeiten mit cron .....                   | 16 - 19       |  |        |
| Crontab für den Benutzer .....                           | 16 - 20       |  |        |
| <b>17 Hilfe zur Selbsthilfe.....</b>                     | <b>17 - 1</b> |  |        |
| Ordnung ist das halbe Leben <b>R</b> Rechner-Ordner..... | 17 - 2        |  |        |
| Kommandos, die Auskunft geben über das System.....       | 17 - 3        |  |        |
| Online-Hilfen: man, info und Co. ....                    | 17 - 4        |  |        |
| SUSE-spezifische Hilfen.....                             | 17 - 5        |  |        |
| Hilfesysteme mit grafischer Oberfläche .....             | 17 - 6        |  |        |
| Hilfe im Internet.....                                   | 17 - 7        |  |        |

# 1 Linux/Unix Grundkonzepte

## 1.1 Betriebssystem - Ein System um einen Rechner in Betrieb zu halten

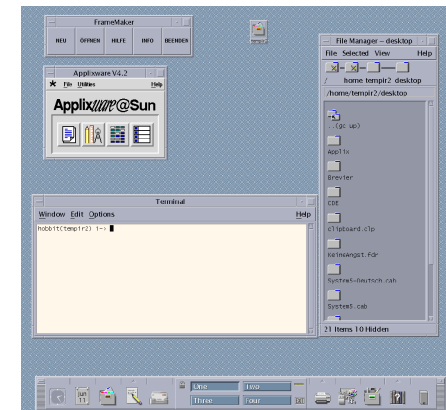
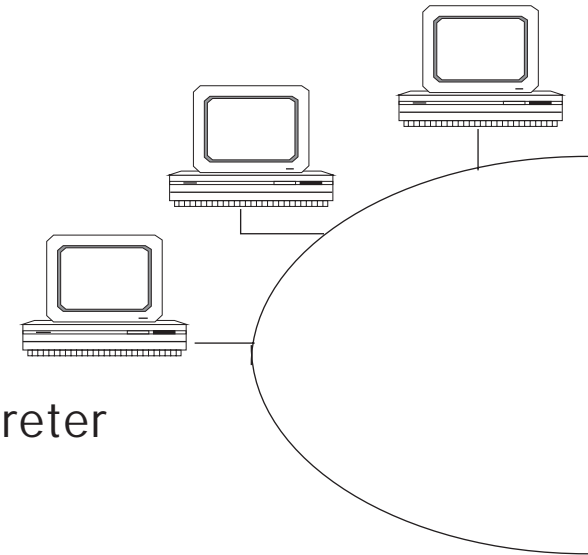


# Betriebssystemarten

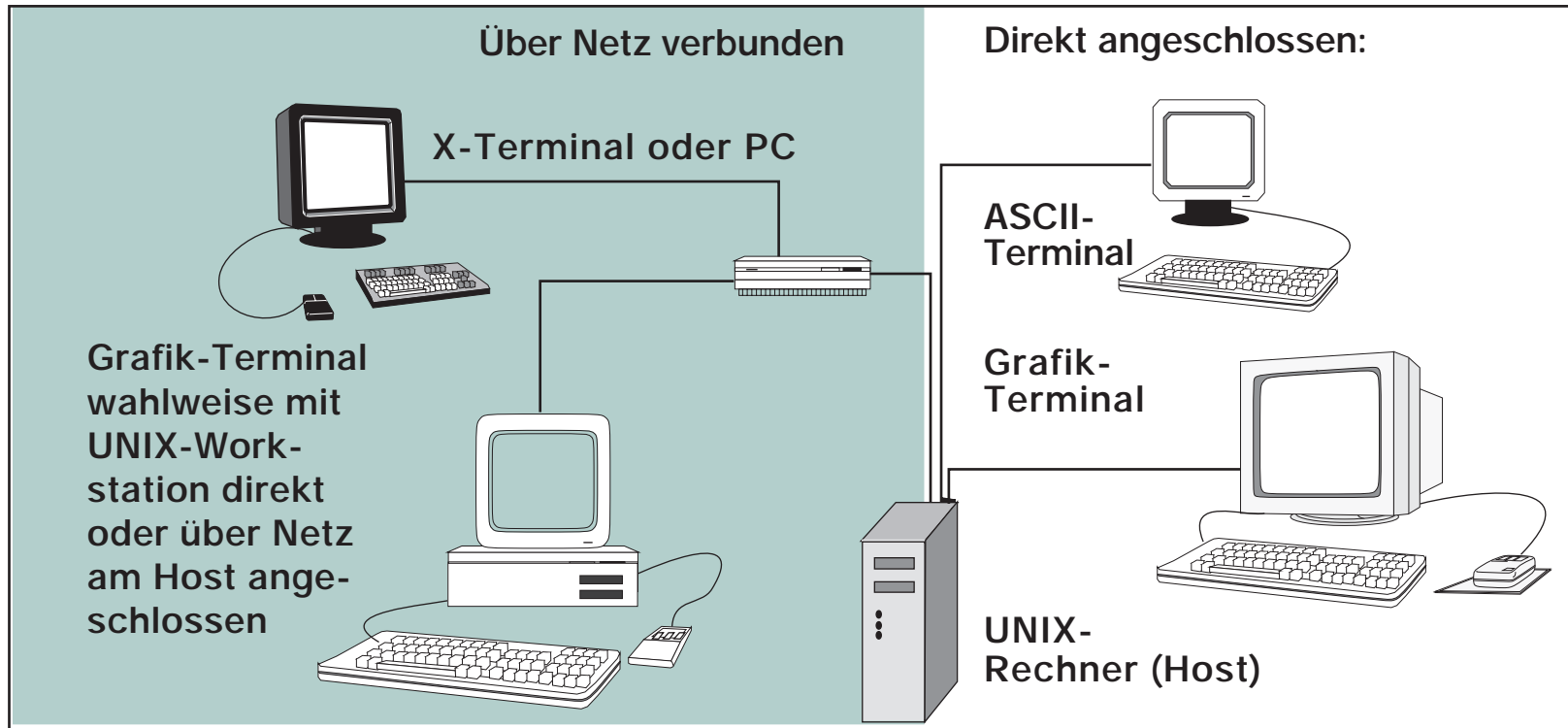
|                       |  |                                 |
|-----------------------|--|---------------------------------|
| <b>Single-Tasking</b> | <b>Single-User</b><br>MS-DOS<br>CP/M<br>...    | <b>Multi-User</b>               |
| <b>Multi-Tasking</b>  | OS/2<br>Windows NT<br>Windows 95<br>MS-Windows | UNIX<br>VM/CMS<br>BS2000<br>VMS |

# Was ist das Besondere an UNIX/LINUX?

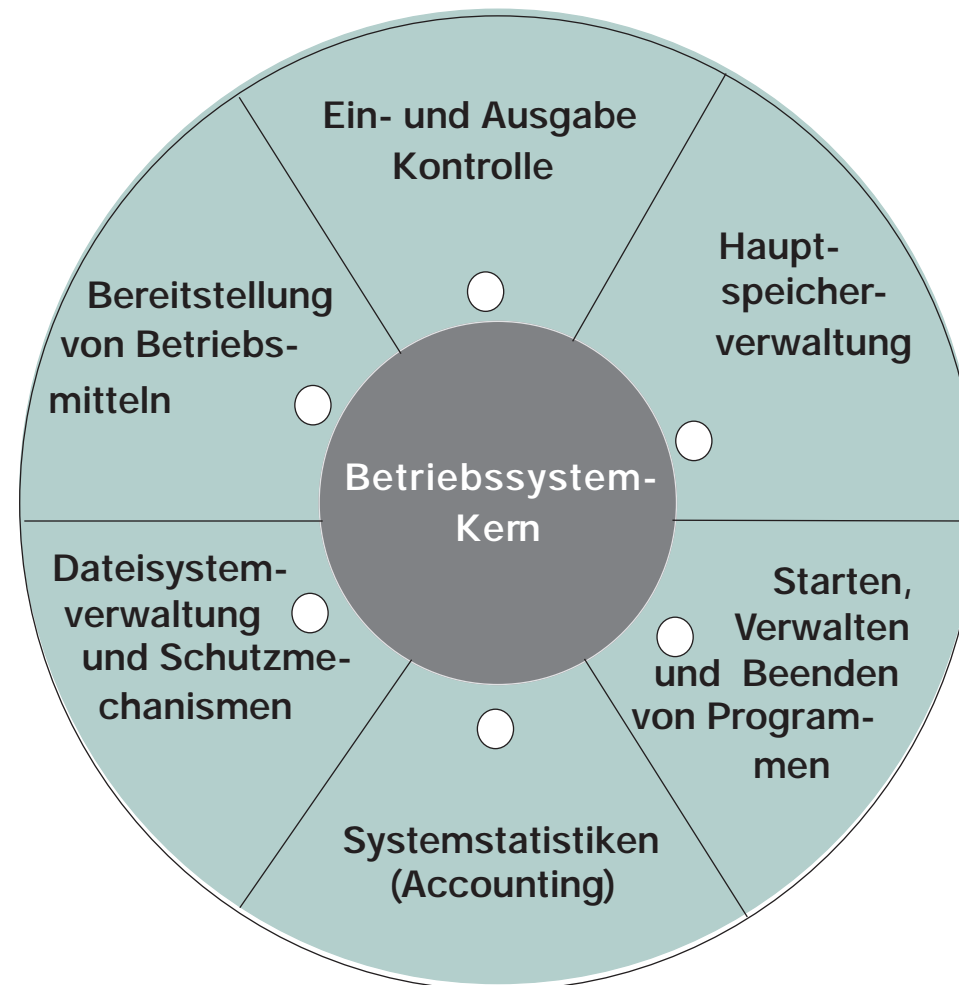
- ❑ Multi-User-Betrieb
- ❑ Multi-Tasking
- ❑ Timesharing
- ❑ Dialogverarbeitung
- ❑ Individuelle **Zugriffsrechte** für Dateien
- ❑ **Shell** – Benutzerschnittstelle und Kommandointerpreter
- ❑ Verfügbarkeit zahlreicher Sprachen
- ❑ Eine Vielzahl von Dienstprogrammen (Software-Tools)
- ❑ **Netzwerkfähigkeit** – Kommunikation
  - **uucp** (*unix to unix copy*)
  - **TCP/IP**  
(*Transmission Control Protocol-Internet Protocol*)
- ❑ **Grafische Benutzeroberfläche** – CDE, KDE, Gnome  
(*Common Desktop Enviroment*)



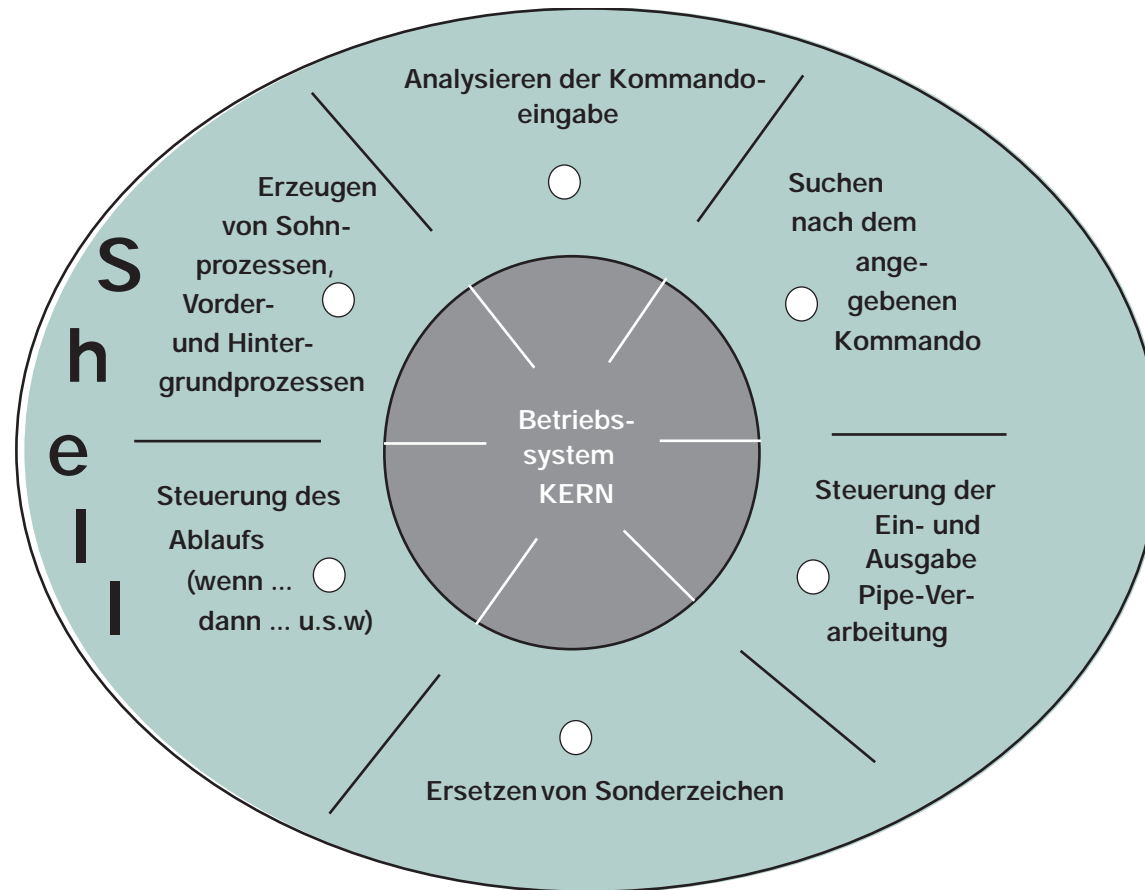
# Multi-User und Multi-Tasking



# Aufgaben des UNIX Betriebssystems



# Einordnen der Shell (Bash)



## Welche Portierungen gibt es von UNIX

| UNIX-Derivat   | Hersteller, Sytemhaus oder Distribu-<br>tor*  | Adresse im Web <a href="http://www">http://<br/>www.</a>                        |
|--|---|---|
| <b>AIX</b>   | IBM   | ibm.de  |
| <b>BSD/OS</b>  | Berkeley Software Design<br>University of California, Berkeley                          | bsd.com   |
| <b>HP-UX</b>   | Hewlett Packard   | hp.com  |
| <b>IRIX</b>  | Silicon Graphics SGI  | sgi.com   |
| <b>LINUX</b>   | GNU (FSF Free Software Foundation)<br>Debian<br>RedHat<br>Slackware<br>SuSE<br>Mandrake | gnu.org<br>debian.com<br>redhat.com<br>slackware.com<br>suse.de<br>mandrake.com |
| <b>Monterey/64</b>                                     | IBM, SCO, Sequent, In tel   | sco.com/monterey  |
| <b>UnixWare</b>  | SCO   | sco.com   |
| <b>SINIX</b>   | Siemens   | siemens.de  |
| <b>SOLARIS</b>   | Sun Microsystems  | sun.de  |
| <b>ULTRIX, OSF/1<br/>Digital UNIX<br/>True 64 UNIX</b> | COMPAC - DEC (HP)   | dec.com oder compaq.com   |

# Einsatzmöglichkeiten von Linux

| Wofür                      | Programme unter Windows                            | Alternative unter Linux                           |
|----------------------------|--|---|
| <b>Office/System</b>       |  |   |
| Textverarbeitung           | MSWord (.doc)                                      | OpenOffice Writer (.sxw)<br>StarWriter (.sxw)     |
| Tabellenkalkulation        | Excel (.xls)                                       | OpenOffice Calc (.sxc)/StarCalc (.sxc)            |
| Präsentation/<br>Zeichnung | PowerPoint (.ppt)                                  | OpenOffice Impress (.sxi)<br>nur Zeichnung (.sxd) |
| Bildverarbeitung           | Adobe Photoshop (.psd) – und alle gängigen Formate | GIMP (.xcf) – und alle gängigen Formate           |
| Scannen                    | Nero, OnCD u. a.                                   | XScanImage  |
| Adressbuch, Kalender       | Outlook  | Evolution, KOrganizer, Kontact                    |
| Desktop-Datenbank          | Access   | -<br>evtl. Adabas für StarOffice                  |
| Textdateien bearbeiten     | Ultraedit/Note Tab, WordPad                        | kate, nedit, KWrite                               |
| Dateien verwalten          | Explorer, Norton Commander                         | Konqueror, Midnight Commander                     |
| <b>Internet</b>            |  |   |
| Surfen                     | Internet Explorer, Mozilla, Opera                  | Konqueror, Mozilla, Firefox, Opera                |

| <b>Wofür</b>                    | <b>Programme unter Windows</b>          | <b>Alternative unter Linux</b>              |
|---------------------------------|---|---|
| E-Mails lesen, versenden        | Outlook (Express), Mozilla Mail, Eudora | KMail, Mozilla Mail, Evolution, Thunderbird |
| Dateien transferieren           | ftp, WSFTP, PSFtp                       | sftp, Konqueror, gFTP/Downloader X          |
| <b>Grafik</b>                   |   |   |
| Bilder betrachten               | Photoshop, Paint Shop Pro               | GIMP  |
| Digicam-Bilder lesen/bearbeiten | Fotofix                                 | gtkam                                       |
| Bilder betrachten               | IrfanView, XnView                       | KuickView, XnView                           |
| <b>Video</b>                    |   |   |
| Video abspielen                 | Windows Media Player                    | mplayer, Xine, VLC                          |
| DVD abspielen                   | PowerDVD, WindDVD                       | Ogle, mplyer, Xine, VLC                     |
| DVD umwandeln                   | DVDy                                    | dvdxrip, Drip                               |
| Video konvertieren              | VirtualDub, TMPGEnc                     | kavi2svcd, Mencoder                         |
| Videoschnitt                    | Adobe Premiere, Studio, MainActor       | MainActor, Kino                             |
| <b>Audio</b>                    |   |   |
| Audio, MP3 abspielen            | WinAmp, Windows Media Player            | XMMS, Zinf                                  |
| CD-Grabber                      | CDex, Exact Audio Copy                  | Grip  |
| Audio aufnehmen/bearbeiten      | SoundForge, Cool Edit                   | Rezound, Sweep                              |
| CDs/DVDs brennen                | WinOnCD, Nero Burning Rom               | K3b, GCombust, Xcdroast                     |

# Unterschiede von Linux zu Windows

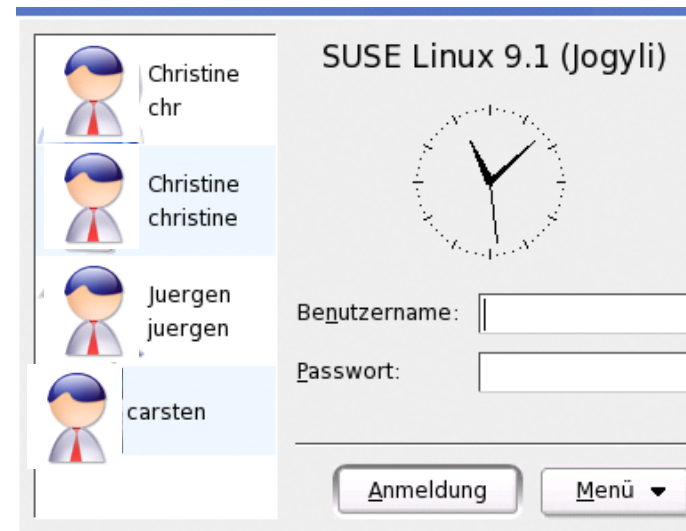
| Begriff                            | Windows (ab NT)   | Linux   |
|------------------------------------|---|---|
| 2 Systemart                        | Single User (Multi User nur mit Terminal-Server)  | Multi User  |
| 3 Ablage der eigenen Dateien       | C:\Dokumente und Einstellungen\ <b>Benutzername</b> \Eigene Dateien   | /home/ <b>Benutzername</b>  |
| 4 Dateinamen<br>Pfade              | Bis 252 Zeichen (nahezu alle Zeichen inklusive Leerzeichen) mit . und 3 Zeichen für Suffix<br><br>Pfadtrennung mit \<br>Dateitypkennzeichnung durch Suffix .nnn | 256 Zeichen<br><b>keine Leer- und Sonderzeichen!</b><br>Groß-/Kleinbuchstaben<br>empfohlen a-z A-Z 0-9<br>und Sonderzeichen - _ .<br>Pfadtrennung mit /<br>Suffix nicht zwingend <sup>2</sup> |
| 5 Ausführbare<br>Dateien/Programme | Gekennzeichnet mit Suffix (.exe, u. a.)   | Durch Zugriffsrecht x   |
| 6 Optionen                         | Bei Kommandoingabe mit /  | Mit -   |
| 7 Laufwerke                        | C:, D: ...<br>Ab Windows NT besteht auch die Möglichkeit, über eine Art mount eine Partition an ein Verzeichnis zu knüpfen.                                     | Eingebunden in den Dateibaum per mount; bzw. beim Hochfahren über /etc/fstab  |

| Begriff          | Windows (ab NT)   | Linux  |
|------------------|---|--|
| 8 Zugriffsrechte | Über Vererbung: Ordner Freigabe, Berechtigungen (Zulassen, Verweigern) für jeden oder bestimmte Benutzer: Vollzugriff, Ändern, Lesen<br>zusätzlich max. Anzahl Benutzer<br>Zugriffseigenschaften von Dateien: schreibgeschützt, versteckt, Archiv | Alle Dateien/Verzeichnisse/Geräte<br>read, write, execute<br>(Lesen, Schreiben, Ausführen) für<br>Besitzer, Gruppe, Andere |
| 9 Sicherheit     | Auch Benutzeranwendungen erfordern oft Administratorrechte<br>Immer wieder Sicherheitslücken, dadurch stärker virenanfällig<br>Ab XP wird auch eine Firewall mitgeliefert   | Streng getrennt nach Administrator (root) und Benutzern<br><br>Mitgelieferte Firewall                                      |

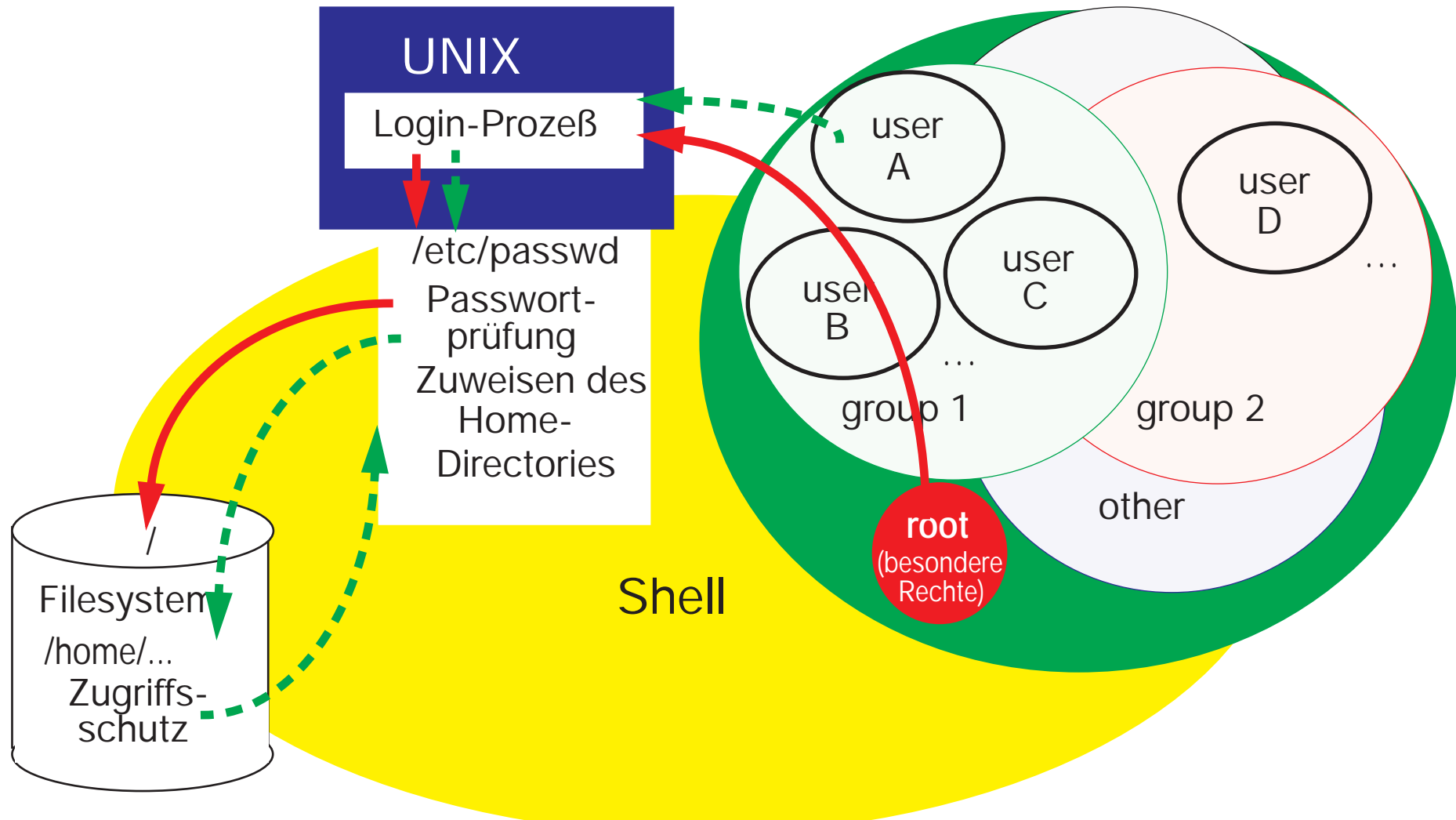
# 2 Auf los geht's los



# Wie melden Sie sich an



# Root und „normale Benutzer“



# Beispiel eines Teilausschnittes der /etc/passwd

|   |   |
|---|---|
| root:dsVLsXfnLk.sl:0:2::/:  | Eintrag für den Systemverwalter (root)  |
| daemon:xxxx:1:1::/:   | Die ersten Zeilen sollten nicht geändert werden -<br>wichtig für die Ausführung von bestimmten Programmen |
| bin:xxxx:2:2::/usr/src:   |   |
| sys:xxxx:3:3::/usr/sys:   |   |
| adm:xxxx:4:4::/usr/adm:   |   |
| ...   |   |
| monika::101:100:Monika Schulze, Tel. 242:/home/kurs/monika:/bin/ksh |   |
| hans::102:100:Hans Mueller, Tel. 440:/home/kurs/hans: /bin/bash     |   |

Angabe für jenes Programm, das nach dem Anmelden gestartet werden soll - z. B. - die verwendete Shell

Login-Directory: Angabe des Verzeichnisses, in das der Benutzer nach dem Anmelden automatisch wechselt (Home-Verzeichnis)

Feld für Kommentar: z. B. Vor- und Nachname des Benutzers

Gruppennummer: Zusätzlich wird in der Datei /etc/group der Benutzer als Mitglied aufgeführt

Identifikationsnummer: Zur Identifizierung aller Dateien/Verzeichnisse des Benutzers

Früher Platz für das verschlüsselte Passwort - nun nur noch ein x als Hinweis für die etc/shadow (Passwortdatei)

Login Name: max 8 Kleinbuchstaben evtl. mit Ziffern - bei Direktzuweisung über useradd auch mehr Zeichen und Großbuchstaben erlaubt.

>< Der Doppelpunkt dient als Trennung der einzelnen Felder (Separator)

# Was bedeuten die Zeichen auf dem Bildschirm?

|    |   |
|----|---|
| \$ | Prompt für den "normalen" Benutzer<br>Bourne- oder Korn-Shell |
| %  | in der C-Shell  |
| #  | Prompt für den Systemadministrator<br>(root)                  |
| >  | Zeichen, daß die Shell noch weitere<br>Eingaben erwartet      |

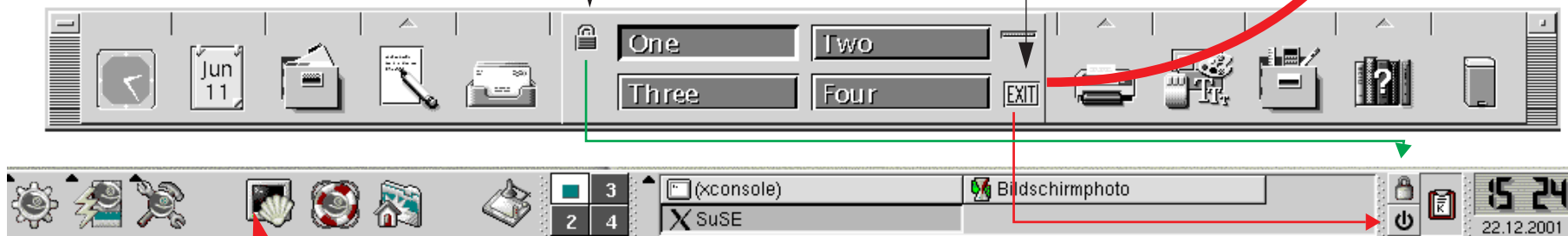
# Abmelden

CTRL + d

exit oder logout

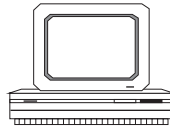
Bildschirm abschließen

Abmelden vom System (EXIT)



UNIX-Befehle werden über ein Terminal eingegeben (es gibt natürlich für viele Befehle auch grafische Programme - doch Systemverwalter sollten die „Kommandoline-Funktion“ kennen!)

## Wie schreiben Sie Kommandos ?



date

Date

DATE

daTE

alle verschieden!

***UNIX kennt nur 'date' !***

# Syntax von UNIX Kommandos

*Kommando [-Option1] [-Option2 ... -Option3] [Parameter1 ... Parameter“n“]*

*oder*

*Kommando [-Option1Option2] [Parameter1 ... Parameter“n“]*

*Beispiel*

entweder:  
oder  
zusammen:

`$ ls -l -a /bin /etc`

`$ ls -la /bin /etc`

Achten Sie auf Leerzeichen

Keine Leerzeichen

**Leerzeichen sind das Kennzeichen zwischen Befehl und Parameter!**

# Einige Kommandos

|        |
|--------|
| date   |
| who    |
| cal    |
| man    |
| passwd |

## Verkettung von Kommandos ";"

*Kommando1 [-Optionen] [Parameter] ; Kommando2[-Optionen] [Parameter]*

### Beispiel

```
$ ls -la /bin /etc ; who ; date
```

*Das Ergebnis der angegebenen Kommandos wird nacheinander auf dem Bildschirm ausgegeben*

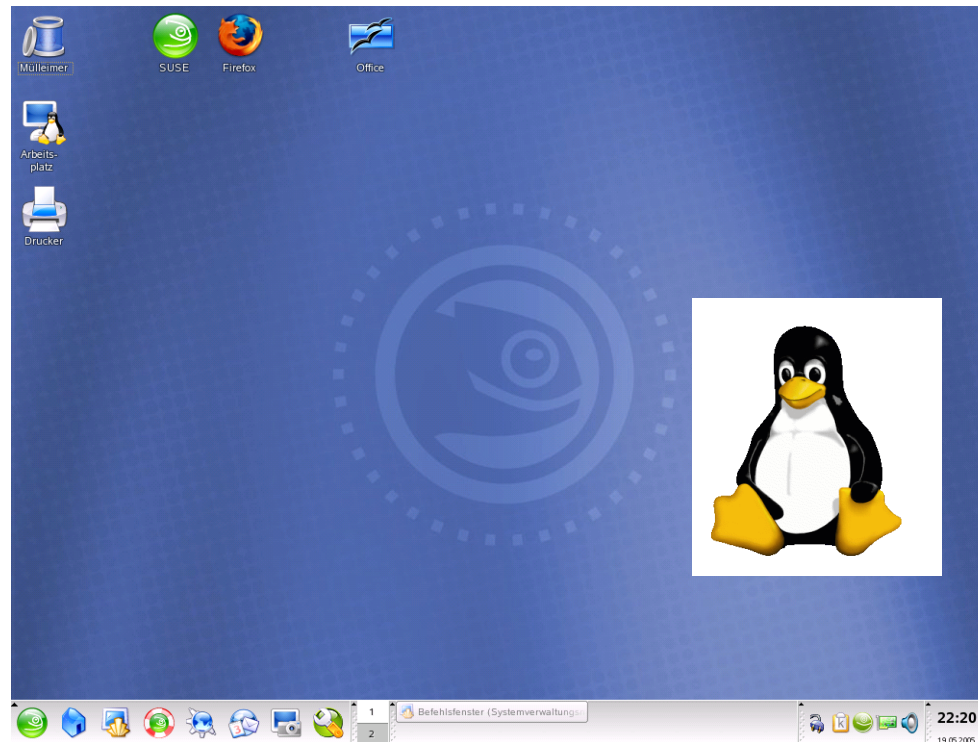
# Zusammenfassung

| Kommandoeingabe   | Funktion   |
|---|--|
| <p><b>date</b> [+ "<i>Formatangaben</i>"]</p> <p><i>Beispiel:</i></p> <p><b>date</b> + "%d.%m. %y"</p><br><p><b>Unter DOS:</b>     <b>date/time</b></p> | <p><b>Zeigt das Datum an</b></p> <p><b>+ Kennzeichen für eigene Formatierung</b></p> <p>%y für Jahr YY<br/>%m für Monat MM<br/>%d für Tag TT<br/>%H - hour 00 - 23<br/>%M - minute 00 - 59<br/>%S - second 00 - 59<br/>%T - time HH:MM:SS<br/>%w - day of week - (Sunday=0)<br/>%h - abgek. Monat - z.B. Jan - Dec</p> |

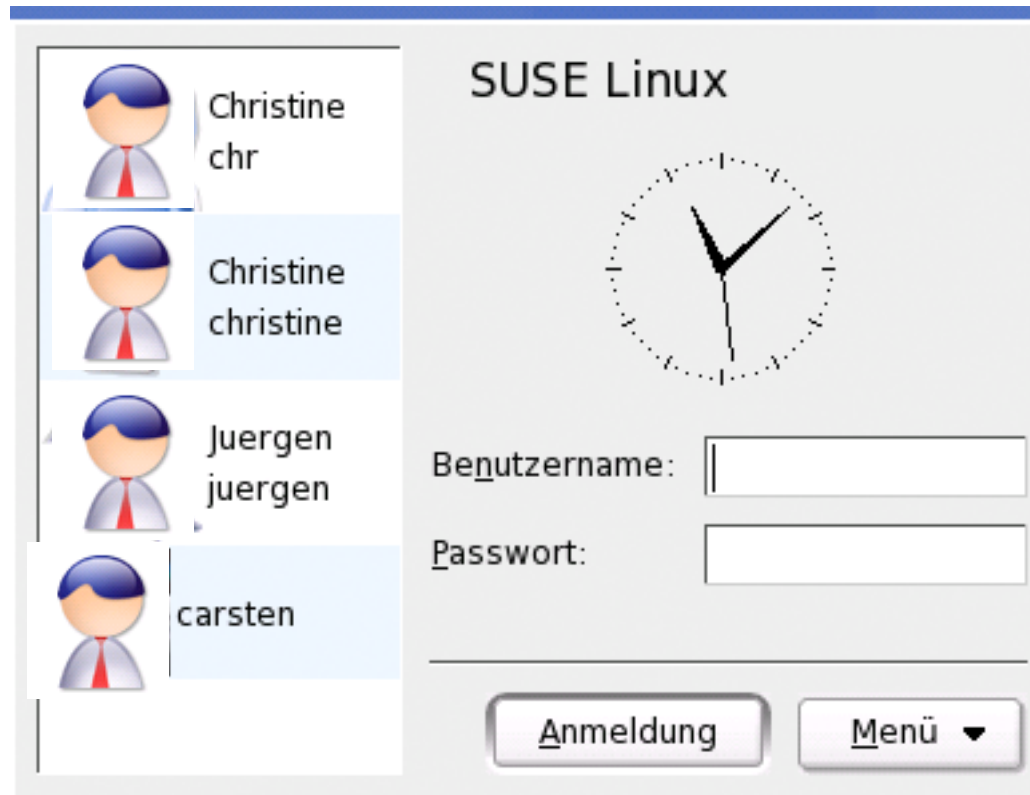
| Kommandoeingabe  | Funktion  |
|--|---|
| <b>who</b><br><b>who am i</b>  | <i>Wer arbeitet am System?</i><br>Zeigt die angemeldeten Benutzer und Terminals<br>Es wird der Benutzernamen, die Terminalbezeichnung und die Anmeldezeit angezeigt |
| <b>cal</b> [[ <i>Monat</i> ] [ <i>Jahr</i> ]]<br><i>Beispiel:</i><br><b>cal 2000   more</b>                | <i>calender</i> -Kalender-Ausgabe<br>Anzeige des aktuellen Kalenders oder vom angegebenen Monat/Jahr  |
| <b>man</b> [ <i>Kapitel</i> ] <i>Kommando</i><br><b>man -k</b> <i>Thema</i><br>Unter DOS: <b>befehl /?</b> | <i>manual</i><br><i>-k Kriterium (Suche nach Thema)</i><br>Gibt eine Beschreibung des angegebenen Kommandos in Verbindung mit more aus                              |

| Kommandoeingabe                     | Funktion  |
|-------------------------------------|---|
| <b>passwd [<i>Benutzername</i>]</b> | Ändert das Paßwort<br>Das Paßwort wird nicht angezeigt und sollte mindestens 6 Zeichen enthalten. |

# 3 Ein kleiner Blick in die grafische Oberfläche KDE

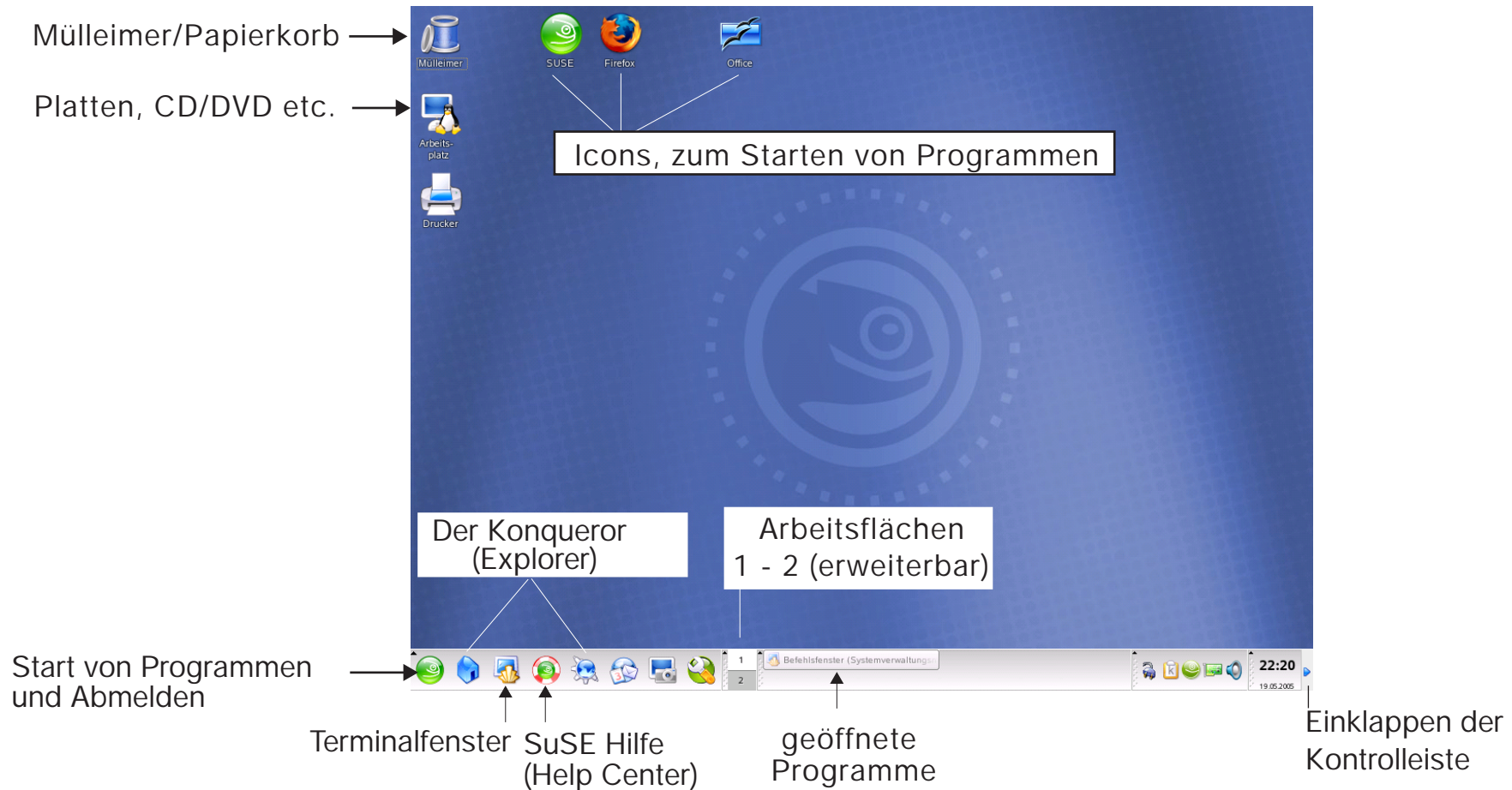


# Wie melden Sie sich an



# Der Desktop - Ihr Arbeitsbereich

- ☐ Wer mit anderen grafischen Oberflächen zurechtkommt, z.B. Windows2000, arbeitet sich auch schnell in das KDE ein



# Wie starten Sie Programme - wo ist was

The image shows the KDE 3.4 desktop environment. On the left is the 'Start of Programs' menu, which is divided into several sections: 'Meistbenutzte Programme' (Most used programs), 'Alle Programme' (All programs), and 'Aktionen' (Actions). The 'Meistbenutzte Programme' section includes 'Kontrollzentrum' (Control Center) and 'Patienten'. The 'Alle Programme' section is categorized by type: 'Spiele' (Games), 'Graphik' (Graphics), 'Internet', 'Multimedia', 'Büroprogramme' (Office programs), 'System', 'Dienstprogramme' (Utilities), 'Dateien suchen' (Search files), 'Kontrollzentrum', 'Persönlicher Ordner' (Home folder), 'SUSE-Hilfe' (SUSE help), and 'Aktionen' (Actions). The 'Aktionen' section includes 'Lesezeichen' (Bookmarks), 'Befehl ausführen...' (Run command...), 'Benutzer wechseln' (Switch user), 'Sitzung sperren' (Lock session), and 'Abmelden...' (Log out...). Below the menu is the 'Start von Programmen' (Start of Programs) icon.

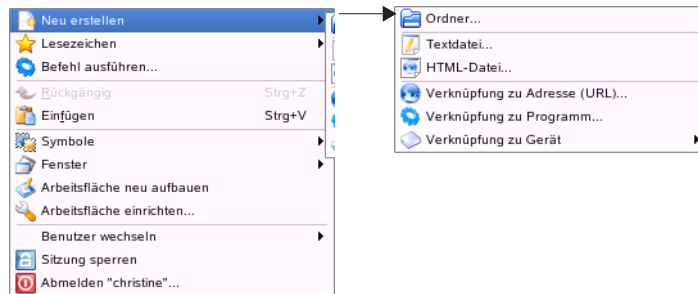
Annotations for the 'Start of Programs' menu:

- 'zuletzt geöffnete Programme' (last opened programs) points to the 'Meistbenutzte Programme' section.
- 'z. B. CD-Wiedergabe' (e.g., CD playback) points to the 'Multimedia' category.
- 'OpenOffice und Dokumentenansicht (Acrobat Reader)' points to the 'Büroprogramme' category.
- 'gute Alternative zum find-Kommando siehe extra Liste' (good alternative to the find command, see extra list) points to 'Dateien suchen'.
- 'Konqueror für das Homeverzeichnis' (Konqueror for the home directory) points to 'Persönlicher Ordner'.
- 'Besonderheit: Zusätzlicher virtueller Bildschirm <Alt+F8> zurück <Alt+F7>' (Special feature: additional virtual screen <Alt+F8> back <Alt+F7>) points to the 'Aktionen' section.

In the center, the 'Desktop' context menu is shown, containing: 'Drucken' (Print), 'Synchronisieren' (Synchronize), 'Archivierung' (Archiving), 'Datei via Bluetooth versenden' (Send file via Bluetooth), 'Editor', 'KGpg', and 'Rechner' (Computer).

On the right, the 'Bonobo-Komponenten-Browser' (Bonobo Component Browser) is shown, listing various system components: 'DMA-Kanäle', 'Ein/Ausgabe-Module', 'Ein/Ausgabe-Ports', 'Geräte', 'Infozentrum', 'Interrupts', 'Netzwerk-Schnittstellen', 'OpenGL', 'PCMCIA', 'Partitionen', 'Prozessor', 'SCSI', 'Samba-Status', 'Sound', 'Speicher', 'Speicher-Geräte', 'Systemüberwachung', 'USB-Geräte', 'X-Server', and 'hp-toolbox'.

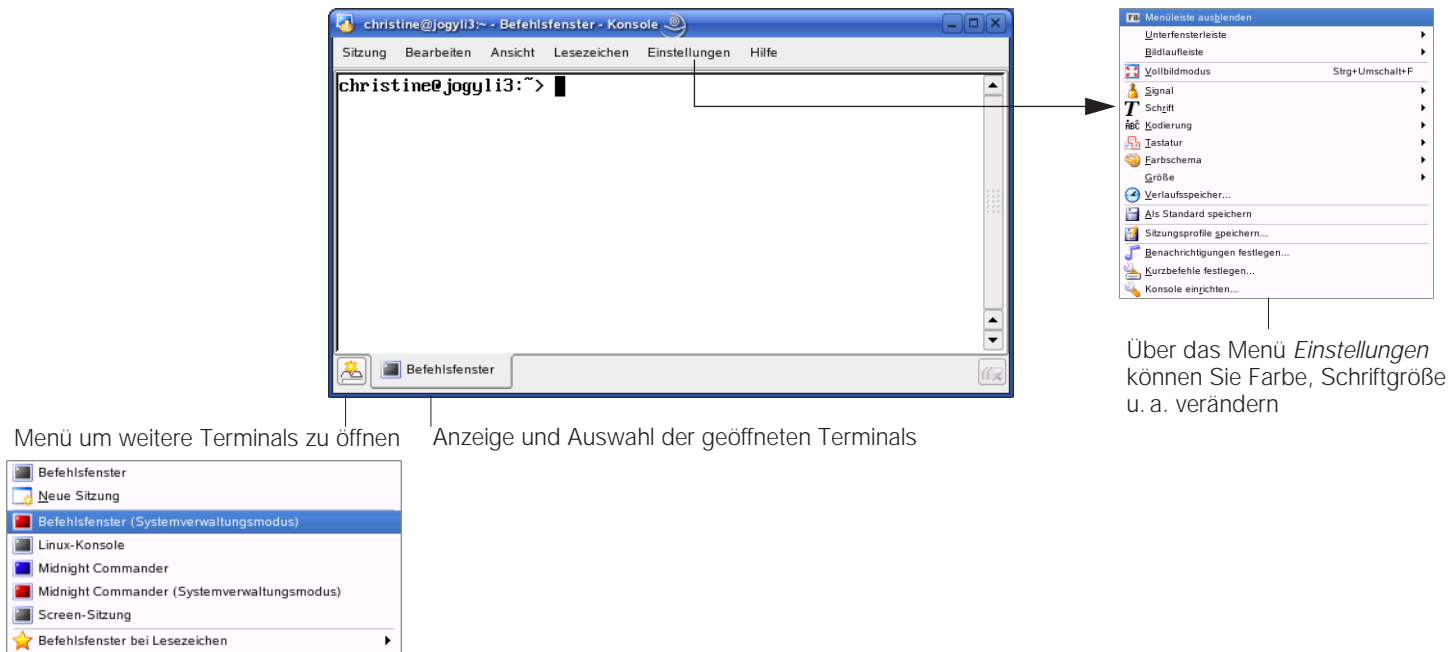
# Kontextmenü auf dem Desktop ...



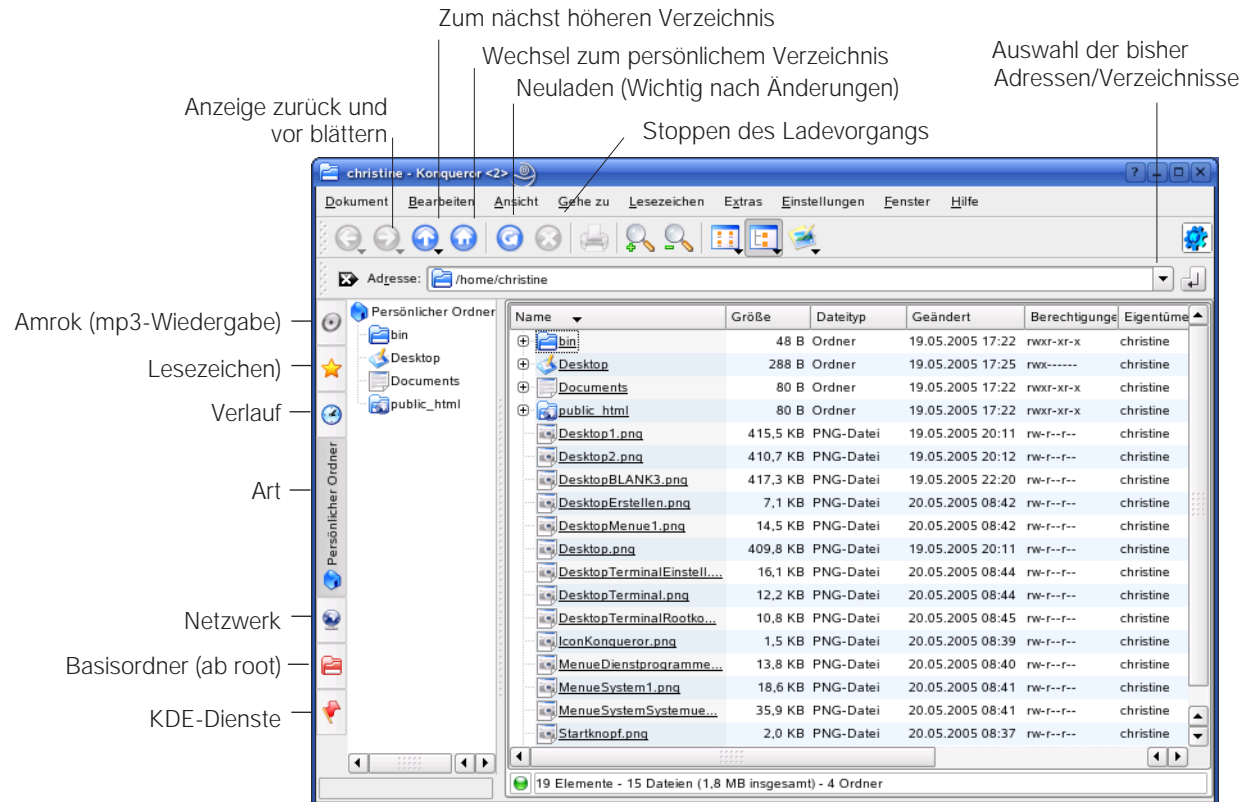
Mit *Neu erstellen* werden Symbole (Icons) auf dem Desktop abgelegt, erst durch weiteres Anklicken wird das damit verbundene Programm gestartet, z. B. bei einer Textdatei der *Erweiterte Editor* (bzw. ab 7.3 je nach Installation z.B. *KWrite*)

# am schnellsten geht's übers Terminal

- ❑ Später einige nützliche Befehle
- ❑ Eigenes Terminal - schnell im Zugriff für die „root“



# Dateiverwaltung mit dem Konqueror



## Grafische Darstellung des persönlichen Verzeichnisses

Bei Anzeigemodus Baumansicht oder Detaillierte Verzeichnisansicht kann nach den unterschiedlichen Rubriken per Mausklick sortiert werden, und die Spalten sind verschiebbar







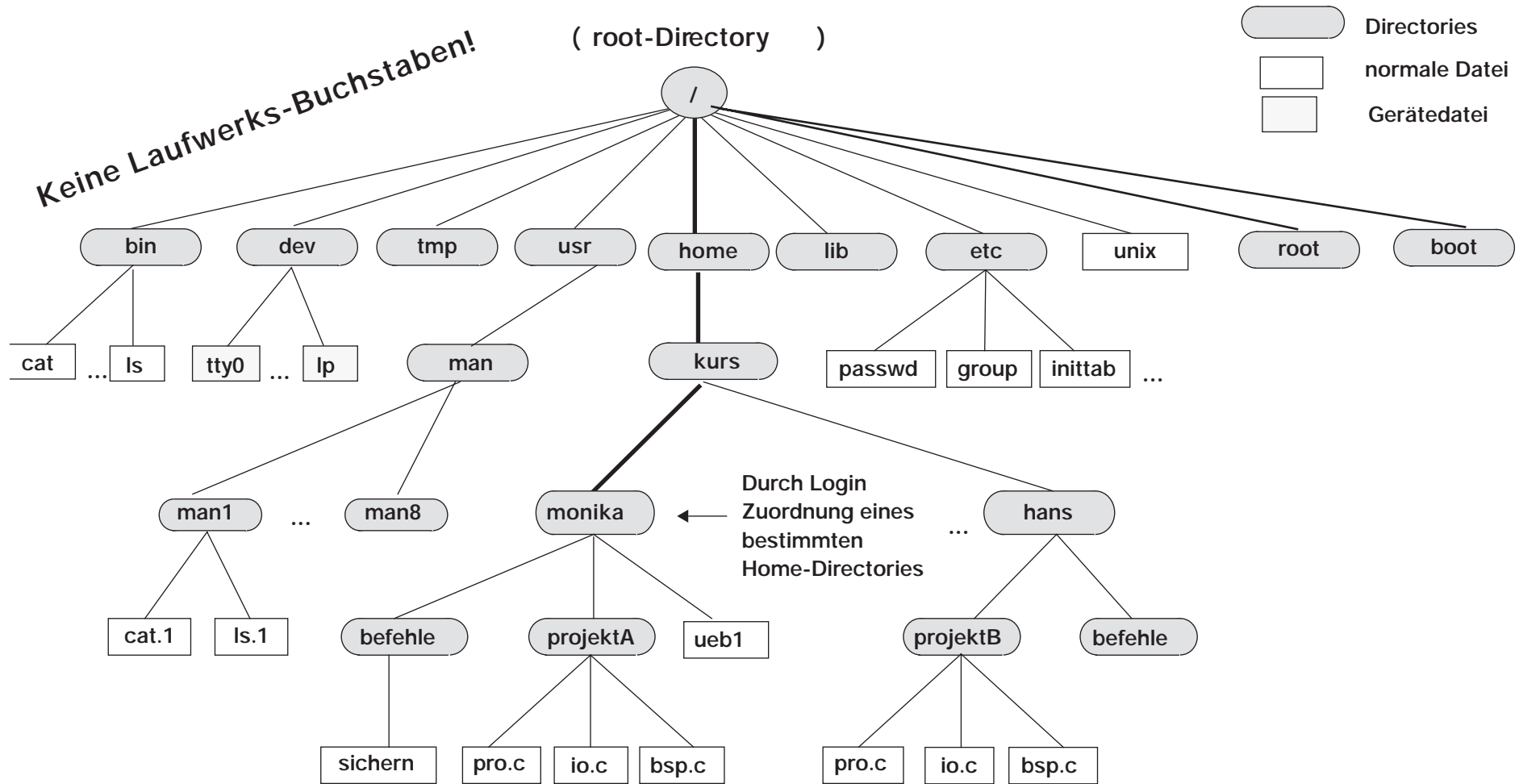








# 4 Das Dateisystem von UNIX





# Absolute und relative Pfadangabe

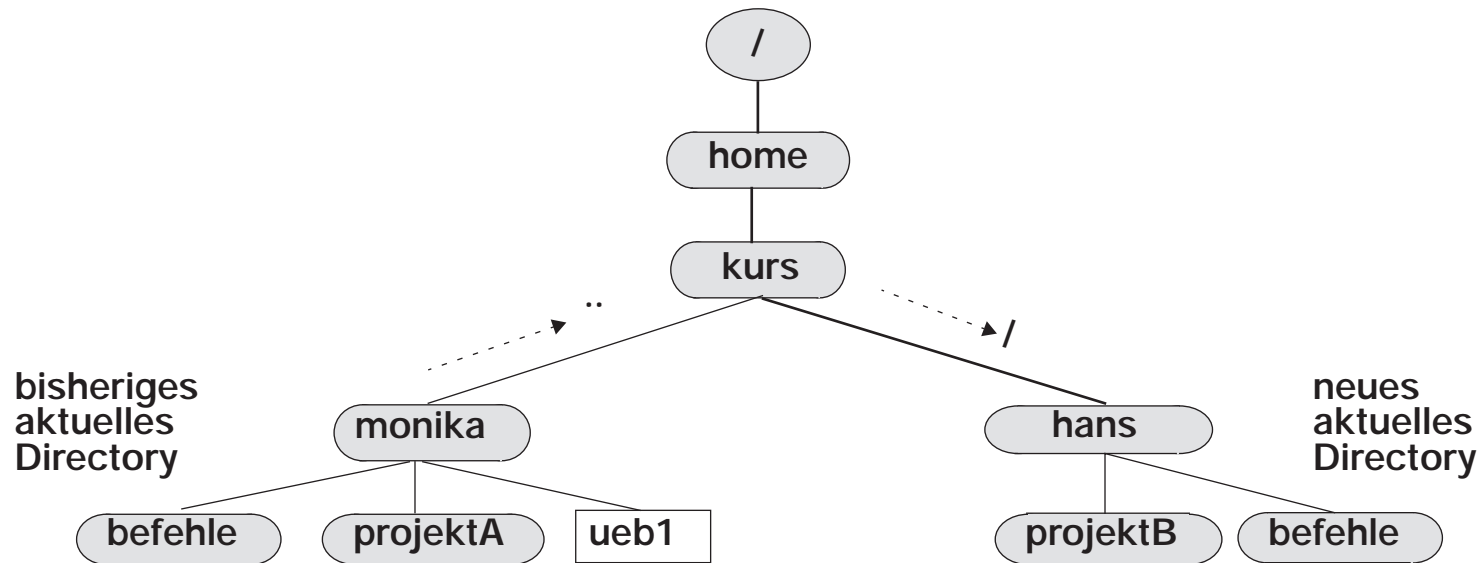
absolute Pfadangabe:

```
$ cd /home/kurs/hans
$ pwd
/home/kurs/hans
$
```

oder

relative Pfadangabe:

```
$ cd ../hans
$ pwd
/home/kurs/hans
$
```



# Wie sehen Sie, wo Sie sind?

**pwd**

(print working directory)

```
$ pwd  
/home/kurs/ben01
```

— Trennung zwischen Directories  
und Dateien

— Kennzeichen für Beginn ab root

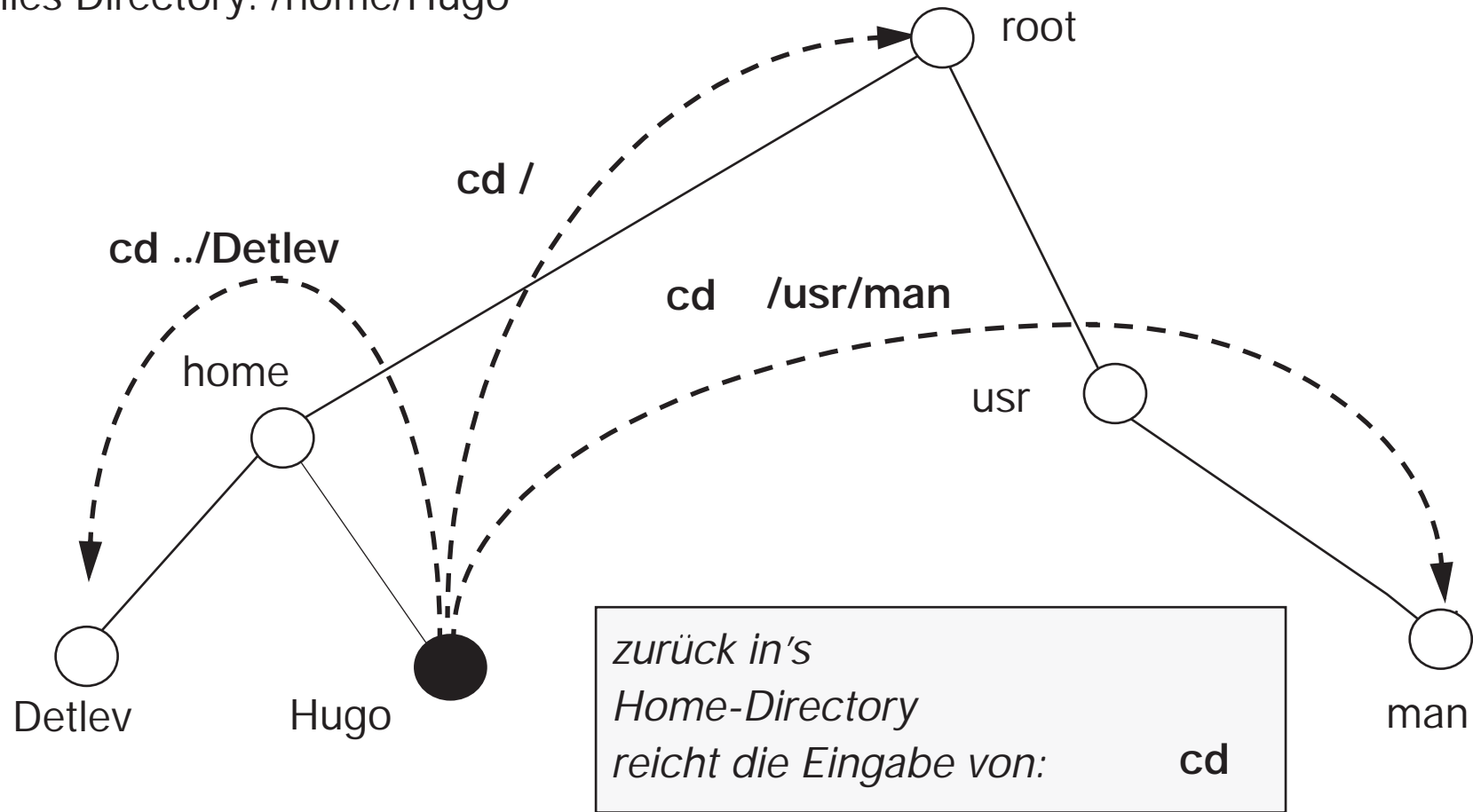
# Dateien und Directories

- ❑ Durch Pfadnamen und Namen der Datei
- ❑ Path/Pfadangaben:  
Folge der Directorynamen, getrennt durch "/"  
absolut: beginnend mit / (root)  
relativ: beginnend im (aktuellen) Directory  
(ohne führenden Trennungs-  
Schrägstrich!)
- ❑ Reservierte Namen:  
/ root des UNIX-Baumes (Wurzel)  
. aktuelles Directory  
.. vom aktuellen Directory das nächst höhere Directory zur root



# Klettern durch den Dateibaum

aktuelles Directory: /home/Hugo



# Kommandos für die Tour

| Kommandoeingabe  | Funktion   |
|--|--|
| <p><b>ls [-FIRda]</b></p> <p>Unter der Bash ls -l gleich mit:<br/> <b>ll</b></p> <p><b>Unter DOS:      dir, tree</b></p> | <p><b>list</b></p> <p><b>Zeigt den Inhalt von Directories bzw. Attribute von Dateien</b></p> <p><b>-F <i>Format short</i></b><br/> <b>Directories sind mit "/" gekennzeichnet, ausführbare Kommandos/Programme mit &gt;*&lt;</b></p> <p><b>-a <i>all</i></b><br/> <b>Zeigt auch versteckte Dateien mit . beginnend</b></p> <p><b>-l <i>long format</i></b><br/> <b>Anzeige mit Attributen</b></p> <p><b>-R <i>Rekursiv</i></b><br/> <b>Der Dateibaum mit sämtlichen Unterdirectories wird angezeigt</b></p> <p><b>-d <i>directory</i></b><br/> <b>Es werden nur die Eigenschaften des Directories angezeigt, nicht dessen Inhalt</b></p> |

| Kommandoeingabe   | Funktion   |
|---|--|
| <b>pwd</b>  | <i>print working directory</i><br><b>Zeigt das aktuelle Directory mit absolutem Pfadnamen an</b>   |
| <b>cd [Directory]</b><br><br><b>Beispiele:</b><br><b>cd</b><br><br><b>cd ..</b><br><b>Unter DOS: cd</b> | <i>change directory</i><br><b>Wechsel in das Home- oder angegebene Directory</b><br><br>Ohne Angabe kehrt man immer ins Home-Directory zurück<br>wechselt in ein Directory nach oben |
| <b>cat Dateiname(n)</b><br><br><b>Unter DOS: type</b>   | <i>concatenate - zusammenfügen</i><br><b>Zeigt den Inhalt von Dateien</b>  |

# Erstellen von Directories?

```
mkdir -p Directory-Name(n)
```

*make directory*

*path* max. 256 Zeichen (vor System V  
noch nicht vorhandene 14 Zeichen) Der Name darf in dem  
Directories werden aktuellen Directory noch nicht  
mit angelegt anderweitig vergeben sein

# Löschen von Directories

```
rmdir Directory
```

*remove directory*

## Weitere Kommandos zur Dateiverwaltung

| Kommandoeingabe  | Funktion   |
|--|--|
| <b>touch</b> <i>Datei(en)</i><br><br><i>Beispiel:</i><br><b>touch</b> <i>neu1 neu2</i> | <i>berühren</i><br><br><b>Aktualisiert das Datum einer Datei bzw. legt sie neu an</b><br><br>Legt die Dateien neu1 und neu2 an |

| Kommandoeingabe  | Funktion  |
|--|---|
| <p><b>cp [-i] <i>Datei1</i> <i>Datei2</i> ... <i>Directory</i></b><br/> <b>cp [-i] <i>Datei1</i> <i>Datei2</i> ... <i>Directory</i></b><br/> <b>cp [-ir ] <i>Directory</i> <i>Directory</i></b></p> <p><b>Unter DOS:      <i>copy, xcopy</i></b></p> | <p><b><i>copy</i></b><br/> <b>Kopiert eine oder mehrere Datei(en) in ein anderes Directory</b></p> <p><b>-i <i>interactive</i></b>      Falls eine Datei mit gleichem Namen schon existiert, wird nachgefragt, ob überschrieben werden darf</p> <p><b>-r <i>recursive</i></b>      Der gesamte Dateibaum wird kopiert</p> |



| Kommandoeingabe   | Funktion   |
|---|--|
| <p><b>mv -i</b> <i>datei1 datei2 ... Directory</i></p> <p><b>Unter DOS:      ren (rename)</b></p>   | <p><i>move, bewegen</i></p> <p><b>Ändert einen Dateinamen oder verschiebt Dateien in ein anderes Directory</b></p> <p><i>-i interactive</i></p> <p>Eine bereits bestehende Datei wird nur dann überschrieben, wenn dies mit y bestätigt wird</p> |
| <p><b>mv</b> <i>Directory Directoryneu</i></p>  | <p>Benennt Directory um</p>  |
| <p><b>mv</b> <i>Directory Directory_vorhanden</i></p> <p>Unter einigen UNIX-Derivaten :</p> <p><b>move</b> <i>Directory Directory_vorhanden</i></p> | <p>Verschiebt Directory mit allen Unterdirectories in ein anderes Directory, das bereits vorhanden sein muß</p> <p><i>Nur unter Linux</i></p>  |

| Kommandoeingabe  | Funktion   |
|--|--|
| <p><b>ln [-s] Originaldatei Linkname</b></p> <p><b>ln -s Original-Directory Link-Directory</b></p> | <p><i>link</i></p> <p><b>Vergibt Dateien zusätzliche Namen bzw. verweist auf eine andere Datei oder ein anderes Directory</b></p> <p>Der lokale Link (ohne -s) kann nur auf Dateien innerhalb der gleichen Plattenpartition erfolgen</p> <p><b>-s symbolic link</b></p> <p>Das Link-Directory soll mit vollem Pfadnamen eingetragen werden.</p> <p>Der symbolische Link wird als eigener Dateityp mit <b>l</b> (klein L - link) gekennzeichnet</p> |

# Suchen von Zeilen innerhalb einer Datei

```
grep [-nilvwh] "Muster" Dateiname(n)
```

*get regular expression*

*header*

der Dateiname wird nicht mit ausgegeben

*word*

Suchmuster muß ein einzelnes Wort sein

*invert*

gibt alle Zeilen aus, die nicht dem Muster entsprechen

*line*

nur die Dateinamen werden angezeigt

*ignore case*

Groß- und Kleinbuchstaben gleichbehandeln

*numerierte Zeilenanzeige* der gefundenen Zeilen  
in einer Datei

# Formatierte Ausgabe

```
pr [-ln -on -wn -n Spaltenanzahl] Dateiname(n)
```

**print**

**1-9**

Der Text wird in die angegebene Anzahl Spalten umgebrochen

*(length)*

Anzahl der Zeilen (Seitenlänge)

*(numbering)*

Die Datei wird mit Zeilennummern ausgedruckt

*(offset)*

Zeicheneinrückung vom linken Rand  
eine Einheit entspricht einem »m«

*(width)*

Anzahl der Zeichen pro Zeile (Seitenbreite)

# Ansehen von Dateiinhalten (less, more oder pg)

```
$ more text1
```

```
Herzlich Willkommen zum UNIX-Einfuehrungskurs
```

```
...
```

```
h
```

```
-----  
h          help  
q or Q     quit  
<blank> or \n next page  
l          next line  
d or ^D    display half a page more  
. or ^L    redisplay current page  
f          skip the next page forward  
n          next file  
p          previous file  
$          last page  
s savelist save current file in savelist  
/pattern/  search forward for pattern  
?pattern? or search backward for pattern  
^pattern^  
!command  execute command  
number    linenummer  
-----
```

Hilfsfunktion

Beenden des Programms  
nächste Seite (Leertaste/Return)  
nächste Zeile  
nächste halbe Seite  
aktuelle Seite nochmal  
überspringe nächste Seite  
nächste Datei  
vorhergehende Datei  
letzte Seite  
sichere aktuelle Datei in savelist  
suche nach dem Muster  
vorwärts  
suche nach dem Muster  
rückwärts  
führe das Kommando aus  
gehe nach Zeile n (n für Zahl)

## Nur ein kleiner Blick in die Dateien:

`head [- n] datei(en)`

*Kopf*

*number*

Anzahl der ersten Zeilen

ohne Angabe werden die ersten 10 Zeilen gezeigt

um die ersten Zeilen einer Datei anzusehen

`tail [-f] [- n] datei(en)`

*Schwanz*

*number*

Anzahl der letzten Zeilen

ohne Angabe werden die letzten 10 Zeilen gezeigt

*following*

Wird die Datei gerade erstellt, können Sie hiermit verfolgen, welche Zeilen hinzukommen

um die letzten Zeilen einer Datei anzusehen

# find – Kommando, um Dateien zu suchen und zu finden

|  |                        |   |   |
|--|------------------------|---|---|
| <b>find</b>  | <i>Start-Directory</i> | [ <i>Suchkriterien/Muster</i> ]   | [ <i>Ausgabeart</i> ]   |
| relativer oder absoluter Pfadname des Directories, von dem ab gesucht werden soll, auch für das aktuelle Directory muß dann der Punkt (.) angegeben werden |                        | <b>-name</b> <i>Dateiname</i><br><b>-user</b> <i>Benutzername</i><br><b>-mtime</b> <i>n (Anzahl der Tage)</i><br><b>-newer</b> <i>Dateiname</i> | <b>-print</b><br>Anzeige der gefundenen Dateien<br><b>-exec</b> <i>Befehl {} ";"</i><br>(ausführen)<br>Ausführung der nachfolgenden Befehle |
| Ohne Angabe von Suchkriterien werden alle Dateien gesucht und mit der Ausgabeart >-print< angezeigt.   |                        |   |   |

## Beispiel von find

```
$ find /home/kurs -name sicher -print
/home/kurs/monika/Befehle/sicher
$ cd
$ pwd
/home/kurs/monika
$ find . -name "s*" -print
./Texte/sprueche
./Befehle/sicher
$ find . -print
./ProjektA
./ProjektA/beispiel.c
./ProjektA/io.c
./ProjektA/pro.c
./Texte
./Texte/kekse
./Texte/sprueche
./Uebungen
./Uebungen/Datum
./Uebungen/inhalt
./Uebungen/neu
./Uebungen/neuer
./Uebungen/ueb1
./Uebungen/datum
./Befehle
./Befehle/sicher
```

— Angabe des gefundenen Dateinamens mit dem absoluten Pfadnamen /home/kurs...

— Suche nach allen Namen mit »s« beginnend  
Angabe der gefundenen Dateinamen mit dem relativen Pfadnamen ab Suchposition ./

— Anzeige aller Dateien mit dem Pfadnamen ab Suchposition, die rekursiv in dem angegebenen Directory gefunden wurden.

## find mit grep kombiniert

```
$ find /home/kurs -user monika -exec grep "denkst$" {} ";"  
/home/kurs/monika/Texte/sprueche: Alles dauert länger als Du denkst
```

Zeigt alle Zeilen von Dateien ab Directory /home/kurs, die monika gehören und bei denen am Zeilenende denkst steht

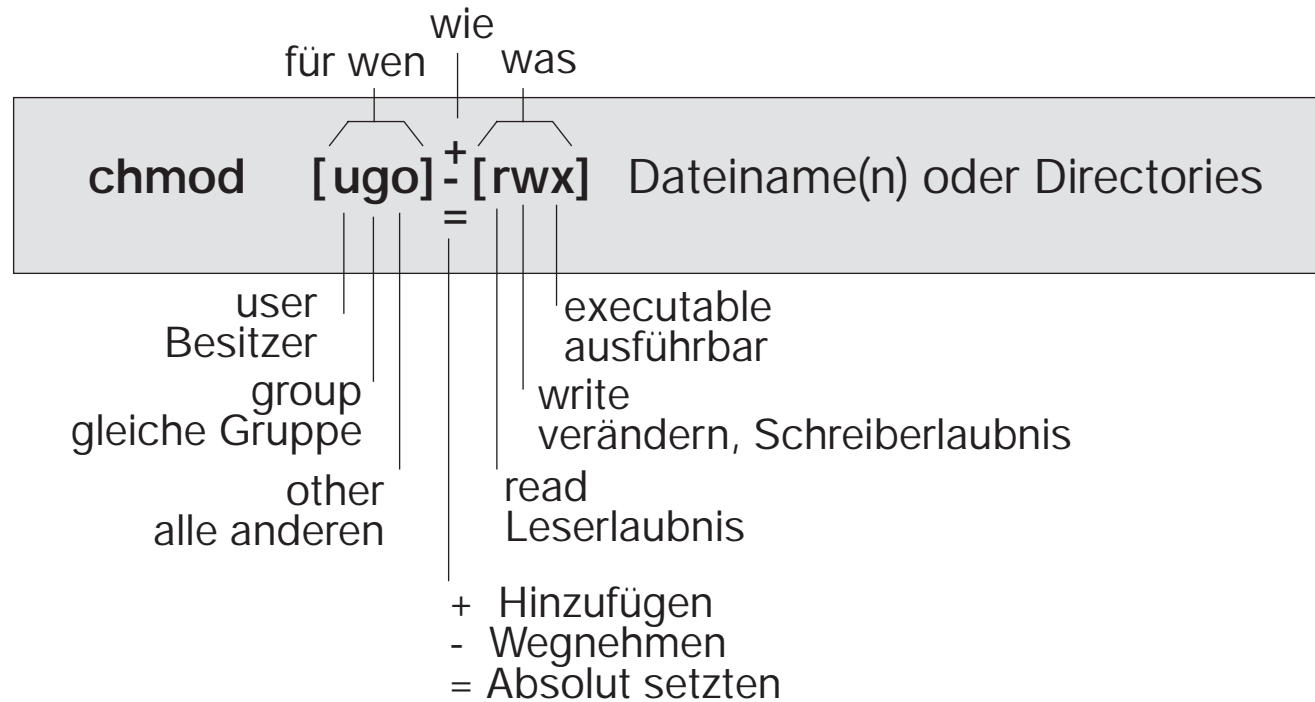
# 5 Zugriffsrechte

|   | Dateityp | Besitzer<br>User | Gruppe<br>Group | Andere<br>Other ... | Name         |
|---|----------|------------------|-----------------|---------------------|--------------|
| normale Datei   | -        | r w -            | r - -           | r - -               | /etc/passwd  |
| Directory   | d        | r w x            | r - x           | r - x               | /bin         |
| Special files<br>(c Character oriented)<br>(b block oriented) | c        | r w -            | - w -           | - w -               | /dev/console |

|   |   |            |   |
|---|---|------------|---|
| r | ≙ | readable   | Leseerlaubnis (4)   |
| w | ≙ | writable   | Schreiberlaubnis (2)  |
| x | ≙ | executable | Ausführbar (1)<br>bei Directories Zugangsrecht( <i>cd</i> ) |

# Verändern von Zugriffsrechten mit Symbolen



# Ändern von Zugriffsrechten mit Oktalzahlen

**chmod** Oktalzahl Dateiname(n) oder Directories

| Zugriffsrechte (Modus) eingeteilt nach: |                           |                             |                        |                           |                             |                        |                           |                             |
|---|---------------------------|-----------------------------|------------------------|---------------------------|-----------------------------|------------------------|---------------------------|-----------------------------|
| Besitzer<br><i>user</i>                 |                           |                             | Gruppe<br><i>group</i> |                           |                             | andere<br><i>other</i> |                           |                             |
| <i>read</i><br>lesen                    | <i>write</i><br>schreiben | <i>execute</i><br>ausführen | <i>read</i><br>lesen   | <i>write</i><br>schreiben | <i>execute</i><br>ausführen | <i>read</i><br>lesen   | <i>write</i><br>schreiben | <i>execute</i><br>ausführen |
| 4                                       | 2                         | 1                           | 4                      | 2                         | 1                           | 4                      | 2                         | 1                           |
| 7                                       |                           |                             | 7                      |                           |                             | 7                      |                           |                             |

# Besitzer- und Gruppenwechsel

**chown [-r]** Besitzername Dateiname(n) oder Directories

**change owner**

Wechsel des Besitzers

ab der Version V.4 gibt es auch die Möglichkeit,  
**rekursiv** alle Dateien eines Directories zu ändern

**chgrp [-r]** Gruppenname Dateiname(n) oder Directories

**change group**

Wechsel der Gruppe

ab der Version V.4 gibt es auch die Möglichkeit,  
**rekursiv** alle Dateien eines Directories zu ändern

## Sonderrechte SUID/SGID-Bit

- ❑ SUID- Bit wird statt x gesetzt. Beim Ausführen des Programmes erhält der Ausführende die Rechte des Programmbesitzers
- ❑ SGID- Bit: Beim Ausführen des Programmes erhält der Ausführende die Rechte der Programmgruppe
- ❑ Beispiel: /usr/bin/passwd (Kommando um das Passwort zu ändern /etc/passwd darf außer der root niemand verändern:

```
-rw-r--r-- 1 root    root        564 Feb 7   /etc/passwd  
-r-s--x--x 1 root    root       12244 Feb 5   /usr/bin/passwd
```

**Kommando um das SUID-Bit zu setzen:**

```
chmod u+s Programm
```

**Kommando um das GUID-Bit zu setzen:**

```
chmod g+s Programm
```

## Sonderrecht Sticky-Bit (t)

- ❑ Dateirechte bleiben erhalten - Verzeichnisrechte werden nicht übergeordnet
- ❑ Beispiel /tmp. Wird dort das t-Bit gesetzt, kann nur dann eine Datei gelöscht werden, wenn auch die Schreibrechte gesetzt sind. (Meist nur der Besitzer), obwohl /tmp Schreibrechte für alle gesetzt hat:

**drwxrwxrwt**

- ❑ Programme mit gesetztem Sticky-Bit werden nicht ausgelagert ) (nicht geswappt)

**Kommando, um das Sticky-Bit zu setzen:**

```
chmod +t Verzeichnis
```

- ❑ Löschen von Dateien nur möglich, wenn mind. eine Bedingung erfüllt ist:
  1. die Datei gehört ihm
  2. er hat w-Recht
  3. das Directory gehört ihm
  4. er ist Superuser

# Oktauzuordnung von SUID/SGID und Sticky-Bit

- Erweiterung der Bereiche um zusätzliche Oktalziffer:

| Sonderrechte |      |                   | Zugriffsrechte ( <i>Modus</i> ) eingeteilt nach: |                           |                                  |                        |                           |                                  |                        |                           |                                  |
|--------------|------|-------------------|--|---------------------------|----------------------------------|------------------------|---------------------------|----------------------------------|------------------------|---------------------------|----------------------------------|
|              |      |                   | Besitzer<br><i>user</i>                          |                           |                                  | Gruppe<br><i>group</i> |                           |                                  | andere<br><i>other</i> |                           |                                  |
| SUID         | GUID | T-Bit<br>(Sticky) | <i>read</i><br>lesen                             | <i>write</i><br>schreiben | <i>eXecute</i><br>ausfüh-<br>ren | <i>read</i><br>lesen   | <i>write</i><br>schreiben | <i>eXecute</i><br>ausfüh-<br>ren | <i>read</i><br>lesen   | <i>write</i><br>schreiben | <i>eXecute</i><br>ausfüh-<br>ren |
| 4            | 2    | 1                 | 4  | 2                         | 1                                | 4                      | 2                         | 1                                | 4                      | 2                         | 1                                |
| 7            |      |                   | 7  |                           |                                  | 7                      |                           |                                  | 7                      |                           |                                  |

Beispiele:

rwsrwxrwt = 1777

rwsr - xr - x = 4755

rwXrWSS - - = 2770

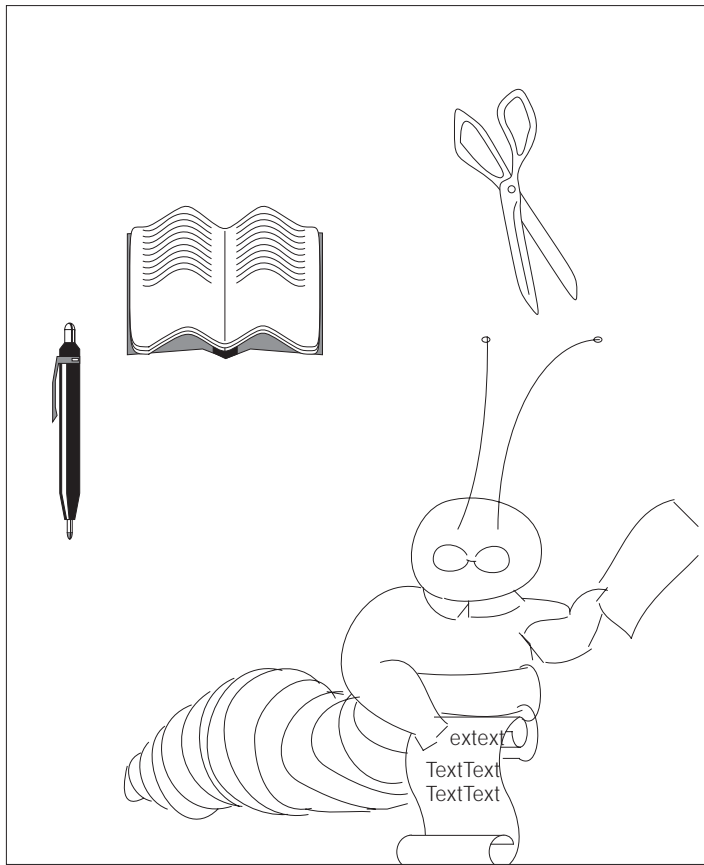
## Voreinstellung über umask

| Eingabe                       | Auswirkung   |             |         |             |          |     |     |                         |       |       |              |            |            |
|-------------------------------|--|-------------|---------|-------------|----------|-----|-----|-------------------------|-------|-------|--------------|------------|------------|
| <code>umask <i>nnn</i></code> | Setzt die Maske für die Zugriffswerte (Minuswert zu Oktalzahl)   |             |         |             |          |     |     |                         |       |       |              |            |            |
|                               | <table><thead><tr><th></th><th>Dateien</th><th>Directories</th></tr></thead><tbody><tr><td>Standard</td><td>666</td><td>777</td></tr><tr><td>z.B. <code>umask</code></td><td>002 -</td><td>002 -</td></tr><tr><td>neuer Modus:</td><td><b>664</b></td><td><b>775</b></td></tr></tbody></table> |             | Dateien | Directories | Standard | 666 | 777 | z.B. <code>umask</code> | 002 - | 002 - | neuer Modus: | <b>664</b> | <b>775</b> |
|                               | Dateien  | Directories |         |             |          |     |     |                         |       |       |              |            |            |
| Standard                      | 666  | 777         |         |             |          |     |     |                         |       |       |              |            |            |
| z.B. <code>umask</code>       | 002 -  | 002 -       |         |             |          |     |     |                         |       |       |              |            |            |
| neuer Modus:                  | <b>664</b>   | <b>775</b>  |         |             |          |     |     |                         |       |       |              |            |            |

Beispiel Teil aus `/etc/profile`:

```
#  
# The user file-creation mask  
#  
umask 022
```

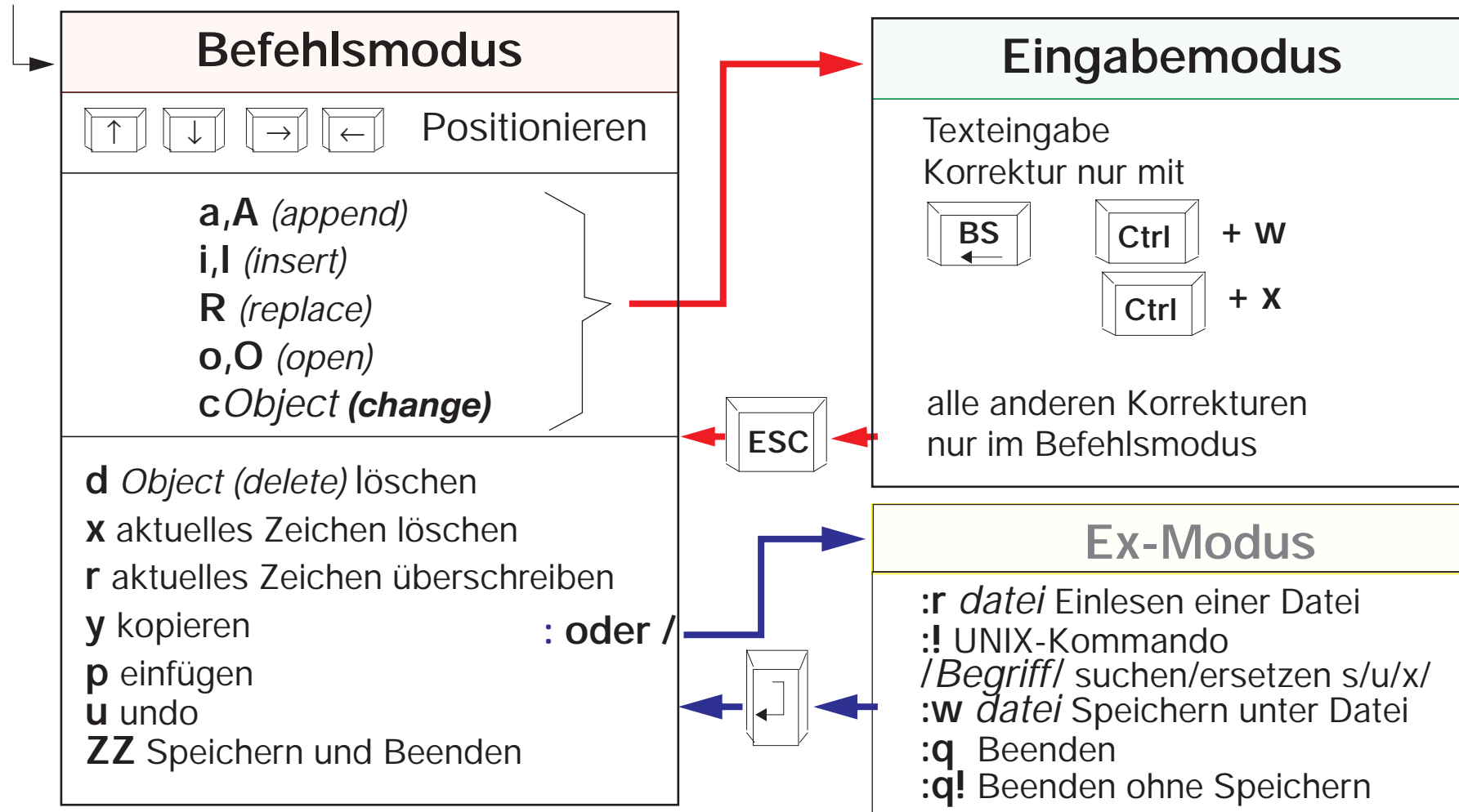
# 6 Der Editor vi und sed



- ❑ Ein mächtiger bildschirmorientierter Editor (*visual interactive*)
- ❑ aber kaum Mnemonik daher schwieriger zu erlernen (old fashion)
- ❑ deshalb nur das Wesentliche, um Dateien zu erstellen und zu korrigieren

# Wie arbeiten Sie mit dem vi?

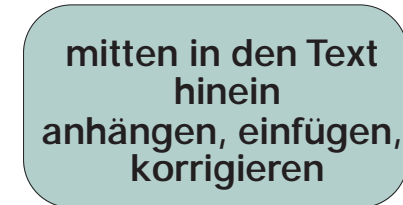
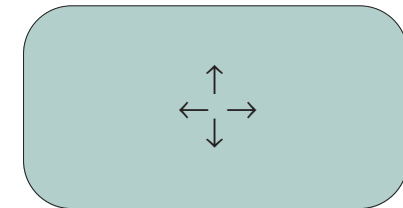
vi datei



# Wie arbeiten Sie mit dem vi?

Sie arbeiten in drei Zuständen

- 1 Kommando-Modus  
um den Cursor zu positionieren  
und Befehle wie löschen, ersetzen, einfügen  
einzugeben
- 2 Eingabe- und Ersetzungs-Modus  
keine Cursortasten (außer im vim)! - Steter Wechsel  
in den Kommandomodus mit
- 3 Eingabe in der Status-Zeile  
für ex-Modus und Suchen:



# Wie arbeiten Sie mit dem vi?

Eröffnen:  
(Der Cursor steht am Anfang der Datei)

**vi** *dateiname*

Anzeige des Textes, Leerzeilen sind gekennzeichnet mit:

~

~

Beenden:

mit Sicherung unter dem aufgerufenen Dateinamen

**ZZ** (nicht sichtbar) schreibt die Datei mit allen evtl. vorgenommenen Änderungen zurück und beendet den vi

Sicherung unter evtl. anderem Namen (*write*):

**:w** *dateiname*

Die Eingabe erfolgt auf der letzten Zeile (Statuszeile) (Exmodus)

und vi beenden (*quite*)

**:q**

ohne Sicherung (vorab nicht :w)

**:q!**

die Originaldatei wird ab letzter Sicherung zurückgeschrieben

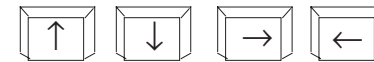
**:e!**

# Wie positionieren Sie den Cursor?

Nur im Kommandomodus! im Zweifelsfalle:  
(im vim auch im Eingabemodes)



Durch die Cursortasten



Zeilenweise durch Return



Zum Zeilenende

\$

Zurück zu Zeilenanfang

^

Wortweise vorwärts (*word*)

w

Wortweise rückwärts (*back*)

b

Zum Anfang der Datei (*1 go*)

1G

Zum Ende der Datei (*go*)

G

Zu Zeilennummer *n*

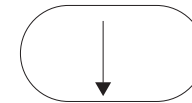
*n*G

# Wie blättern Sie in der Datei?

Bildschirm vorwärts



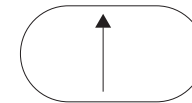
**f**



Bildschirm rückwärts



**b**



# Wie geben Sie Text ein?

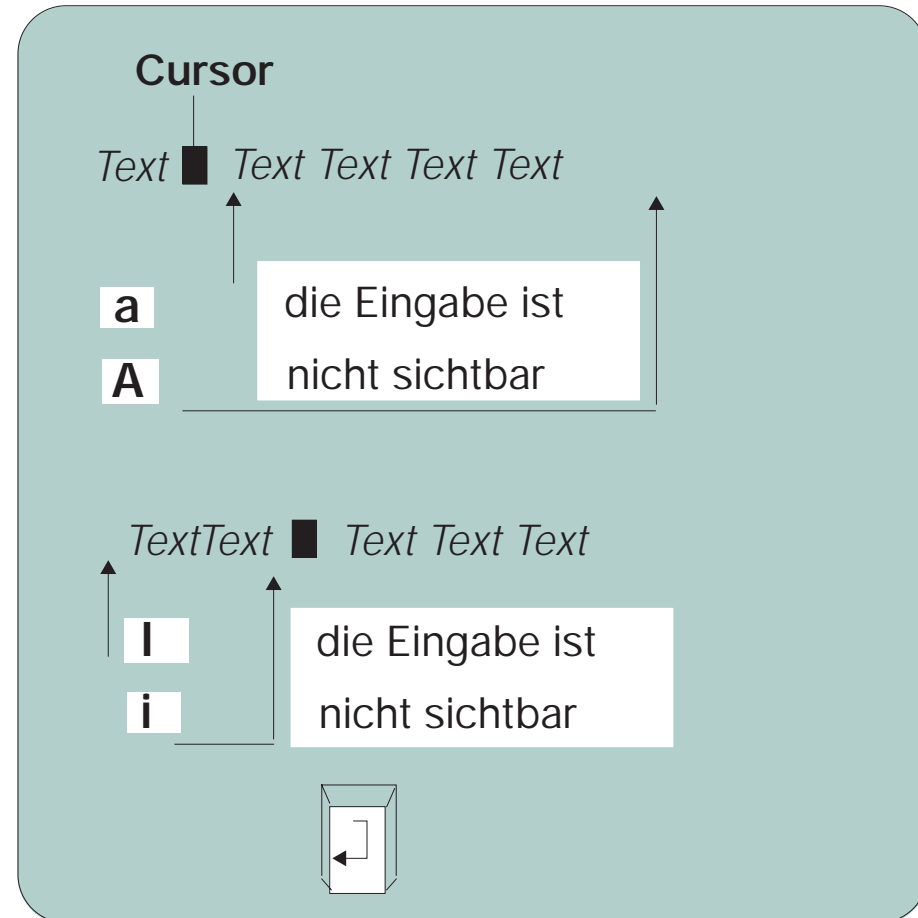
Nur im Eingabemodus:

Durch Anhängen (*append*)

nach dem Cursor  
am Ende der Zeile

Durch Einfügen (*insert*)

am Zeilenanfang  
vor dem Cursor  
neue Zeile durch Return



# Korrekturen im Eingabemodus

(außer bei VIM - Linux)

Löscht das zuletzt  
geschriebene Zeichen

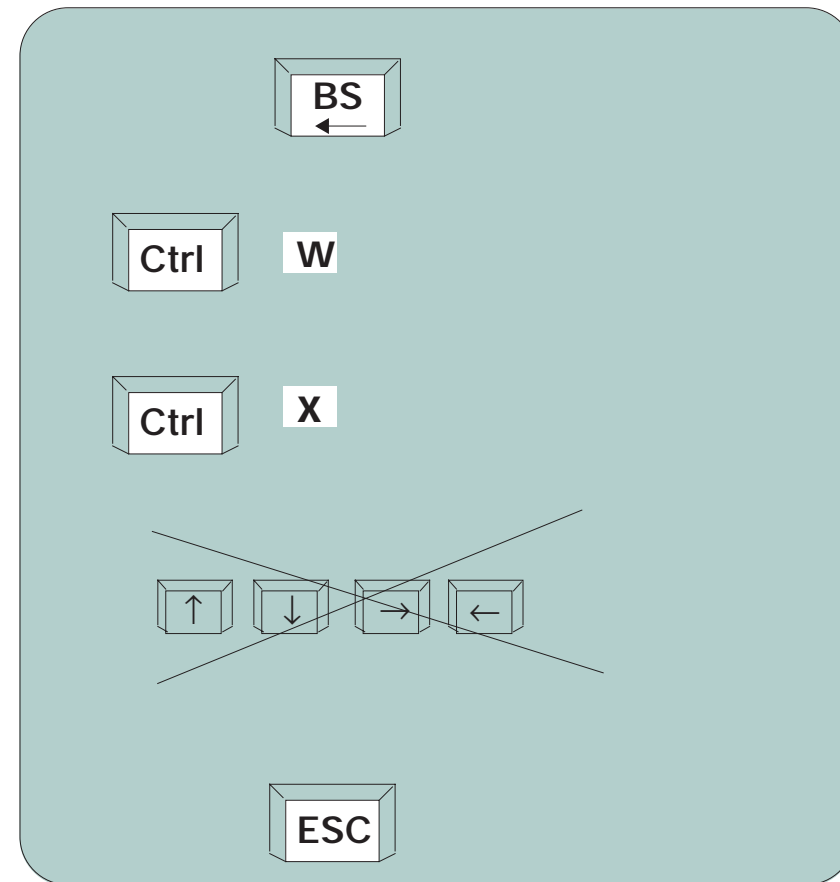
Löscht das zuletzt  
geschriebene Wort

Löscht die  
aktuelle Zeile

Im Eingabemodus keine  
Cursortasten !  
**(im vim auch im Eingabemodes)**

Sonst:

Zum Korrigieren zurück  
in den Befehlsmodus



# Wie korrigieren Sie im Befehlsmodus

Im Zweifelsfalle umschalten:

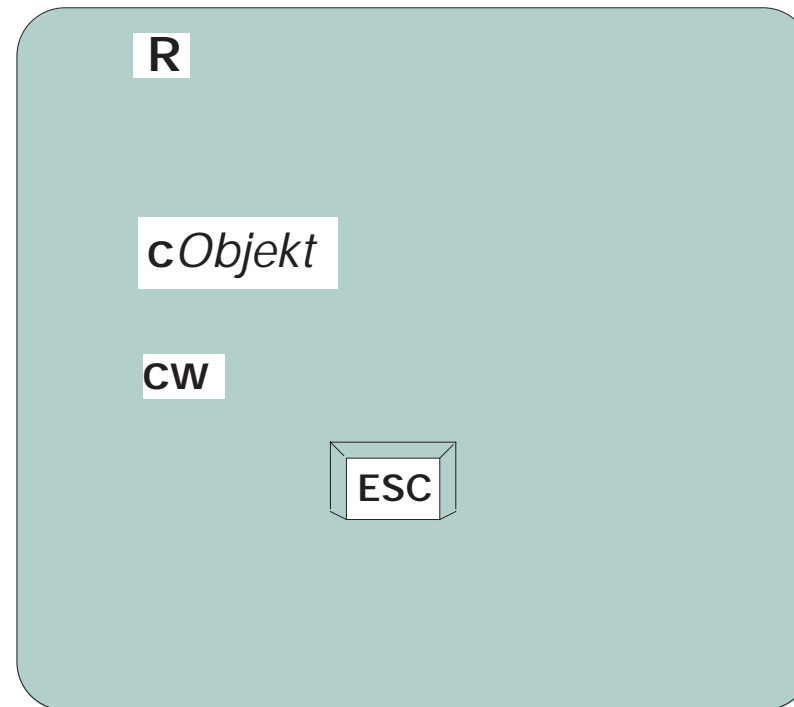


Durch Überschreiben  
(*replace*) von bestehen-  
dem Text ab Cursor

Durch Ersetzen (*change*)  
mit Angabe des Objektes

z.B. nachfolgendes Wort

Abschluß der Eingabe



# Wie löschen Sie im Befehlsmodus?

Cursor auf Fehler positionieren

aktuelles Zeichen (*durch-x-en*)

nachfolgendes Wort (*delete*)

bis zum Ende der Zeile

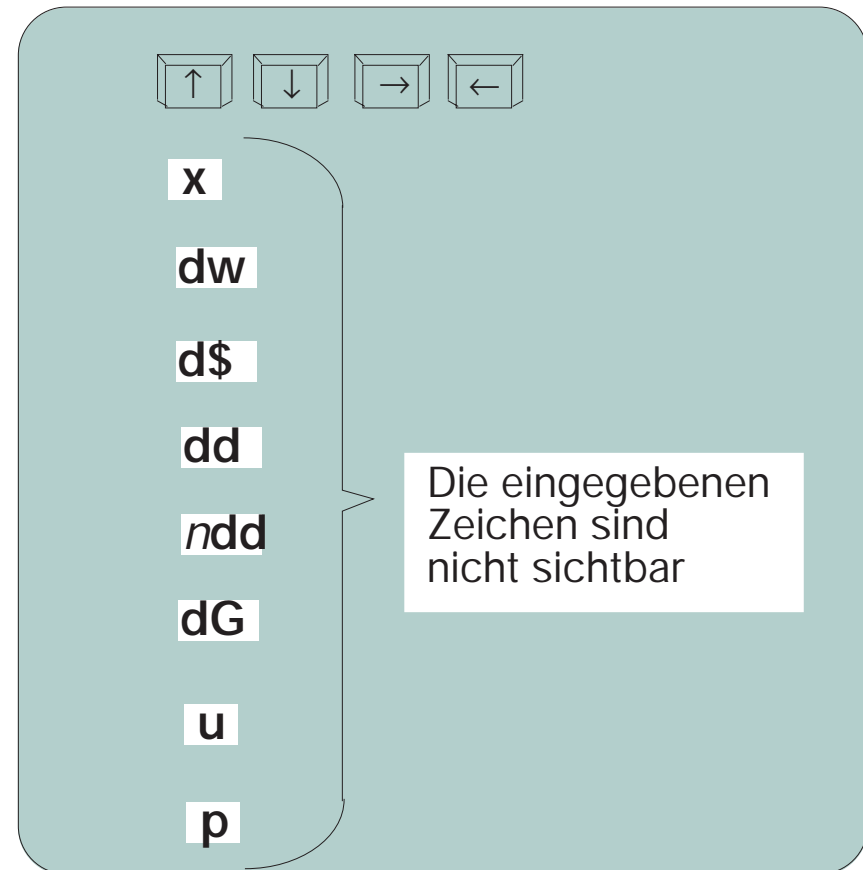
aktuelle Zeile (*delete*)

*n* Zeilen

bis zum Ende der Datei

Letzen Befehl rückgängig machen (*undo*)

Letzte Löschung wieder einfügen (*paste*)



# Wie kopieren und fügen Sie ein?

Im Befehlsmodus Im Zweifelsfalle umschalten



Speichern der aktuellen Zeile

Speichern eines Objektes

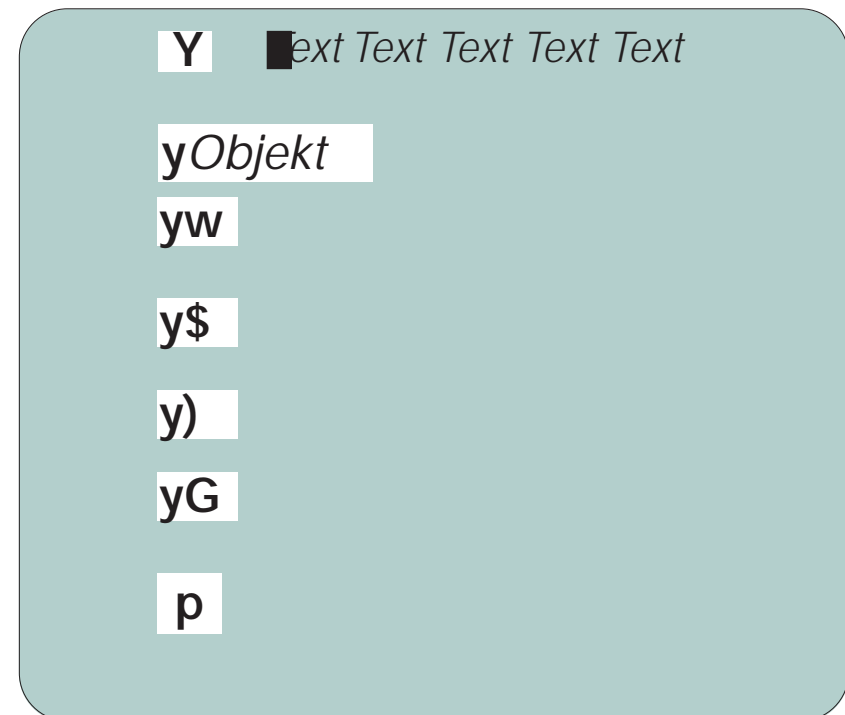
z.B. das aktuelle Wort


bis zum Zeilenende

bis zum Satzende

bis zum Ende der Datei

Einfügen der letzten Speicherung nach der aktuellen Zeile (*paste*)

A light blue rounded rectangle containing a list of vi editor commands. Each command is shown in a white box. The first command is 'Y' followed by a cursor and the text 'ext Text Text Text Text'. The other commands are 'yObjekt', 'yw', 'y\$', 'y)', 'yG', and 'p'.

**Y**  *ext Text Text Text Text*

**y***Objekt*

**yw**

**y\$**

**y)**

**yG**

**p**

# Wie suchen Sie nach Begriffen?

In der aktuellen Zeile (*find*)  
ein Zeichen "x" - vorwärts  
- rückwärts

Wiederholung des letzten  
Suchvorgangs - vorwärts  
- rückwärts

Suchen in der gesamten Datei  
- vorwärts  
- rückwärts

Wiederholung des letzten  
Suchvorgangs - vorwärts  
- rückwärts

**f**x

**F**x

;

,

*/Suchbegriff*

*?Suchbegriff*

/ oder n

? oder N

## Einige im vi verwendete Kürzel

| Zeichen                                      | Bedeutung              | Beispiel              |   |
|--|------------------------|-----------------------|---|
| <b>w</b>                                     | Wort                   | <b>cw</b>             | kopiere das Wort                                |
| <b>\$</b>                                    | bis zum Ende der Zeile | <b>d\$</b>            | lösche bis zum Ende der Zeile                   |
| <i>Befehl doppelt oder in Großbuchstaben</i> | line                   | <b>dd</b><br><b>Y</b> | lösche aktuelle Zeile<br>kopiere aktuelle Zeile |
| <b>)</b>                                     | Satz (bis zum „.“)     | <b>y)</b>             | kopiere den Satz                                |
| <b>G</b>                                     | bis zum Ende der Datei | <b>yG</b>             | kopiere bis zum Ende der Datei                  |

# Sonstige Hilfen

Zusammenfügen von zwei Zeilen

Neuaufbereiten des Bildschirms

Umschalten in den ex-Modus

z.B. Suchen und Ersetzen

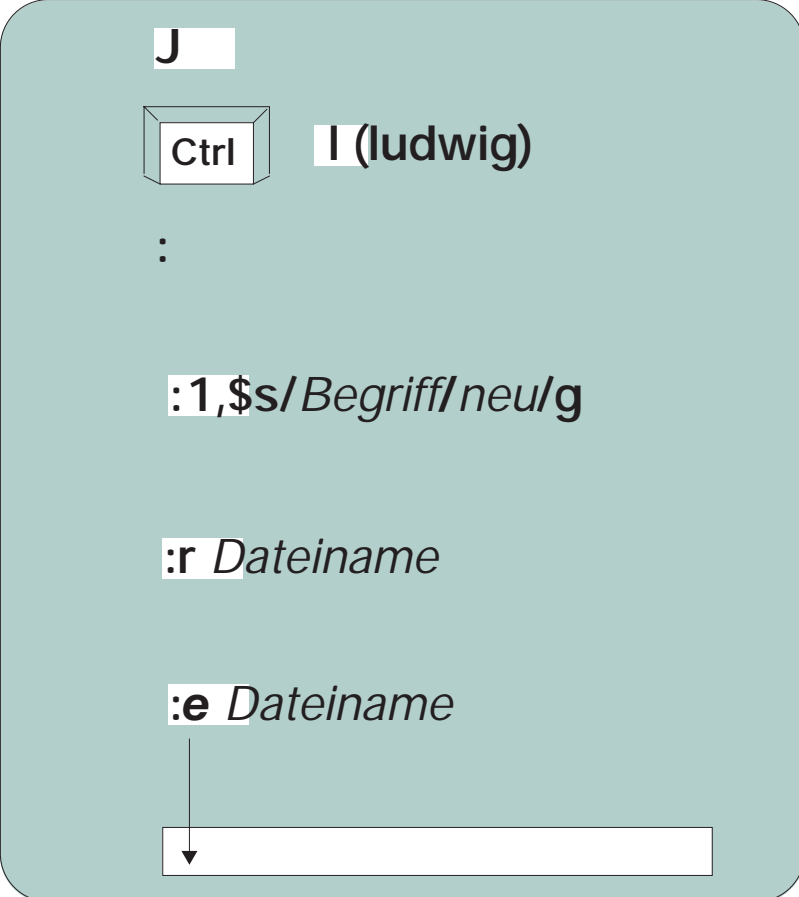
in der gesamten Datei (*global*)

reguläre Ausdrücke wie `/^[^a-x]$/`

Einlesen einer weiteren Datei nach  
der Cursorzeile (*read*)

Editieren einer weiteren Datei

Im ex-Modus (:) erfolgt die Eingabe  
in der Statuszeile



The diagram illustrates the vi editor's status line and command input. It shows a light blue rounded rectangle containing the following elements from top to bottom:

- The letter **J** in a white box.
- A 3D box labeled **Ctrl** followed by the letter **I** and the text **(ludwig)**.
- A colon **:**.
- The command **:1,\$s/Begriff/neu/g**.
- The command **:r Dateiname**.
- The command **:e Dateiname**.
- A white rectangular box with a downward-pointing arrow, representing the status line where the user enters commands.

# Einige praktische Optionen für den vi

(Im vim (Linux) viele schon voreingestellt)

| Kommando             | Bedeutung  |
|----------------------|--|
| <b>:set redraw</b>   | Änderungen werden am Bildschirm sofort nachvollzogen.<br>Ausnahmen sind Korrekturen im Eingabemodus - meist als Default eingestellt<br>Setzt die obige Funktion zurück |
| <b>:set nore</b>     |  |
| <b>:set nu</b>       | <i>number</i><br>Die Datei wird mit laufender Zeilennummer angezeigt<br>Setzt die obige Funktion zurück  |
| <b>:set nonu</b>     |  |
| <b>:set showmode</b> | Hiermit wird in der letzten Bildschirmzeile beim Eingabemodus der Hinweis „Input Mode“ angezeigt   |
| <b>:set all</b>      | Zeigt alle eingestellten Parameter an  |

- Um diese Optionen permanent zu setzen: Eintrag in \$HOME/.exrc  
Mehrere Angaben in einer Zeile sind möglich z. B.  
set number showmode

# SED

## Streamoriented EDitor

```
sed [-n] ' script ' [-f script-Datei] Quelldatei(en) [ > Ausgabedatei ]
```

- ❑ Stream **E**ditor (Datenstrom) liest zeilenweise
- ❑ Schnelles Editieren von Zeichenketten und/oder Dateien
- ❑ Ausgabe nicht in Originaldatei (Quelldatei)
- ❑ Standardausgabe Terminal - bzw. Umleitung in neue Datei
- ❑ Verwendet reguläre Ausdrücke (s. a. grep, vi, more etc.)
- ❑ Optionen:
  - n **keine Bildschirmausgabe**
  - f **nachfolgend Skriptdatei**  
**wird das Skript gleich in die Befehlszeile aufgenommen**  
**unbedingt mit ' .... '**

# Funktionen innerhalb des Scripts

| Funktion                   | Beispiel  | Bedeutung  |
|----------------------------|---|--|
| <b>p</b> <i>print</i>      | <code>sed -n '1,3p' /etc/passwd</code><br><code>sed -n '3,\$p' /etc/passwd</code><br><code>sed -n '/[Mm][ae][iy]*/p' adr</code> | Ausgabe von Zeilen<br>die Zeilen 1-3 werden ausgegeben<br>Ausgabe von Zeile 3 bis Ende der Datei<br>Ausgabe aller Zeilen mit dem Suchmuster /.. /<br>meyer, Meyer, Maier, maier usw. |
| <b>a</b> <i>append</i>     | <code>who   sed '/^schul/a\<br/>Name Terminal LoginZeit'</code>   | Anhängen von Zeilen<br>Nach jedem Benutzer mit schul wird eine Zeile<br>angehängt  |
| <b>i</b> <i>insert</i>     | <code>who   sed '/^schul/i\<br/>Name Terminal LoginZeit'</code>   | Einfügen von Zeilen<br>Vor jedem Benutzer mit schul wird eine Zeile einge-<br>fügt   |
| <b>c</b> <i>change</i>     | <code>ls -l   sed /Total/c\;Gesamt'</code>  | Ersetzen von Zeilen  |
| <b>d</b> <i>delete</i>     | <code>ls -l   sed /Total/d</code><br><code>sed '/^\$/d Datei &gt; Dateineu</code>   | Löschen von Zeilen<br>Entfernt leere Zeilen in einer Datei   |
| <b>s</b> <i>substitute</i> | <code>who   sed 's/ *.*//g'</code>  | Ersetzen von Zeichen<br>g <i>global</i> am Zeilenende: Ersetzung mehrfach pro<br>Zeile (Ausgabe nur Benutzernamen)   |

# Reguläre Ausdrücke - Metazeichen/Wildcards

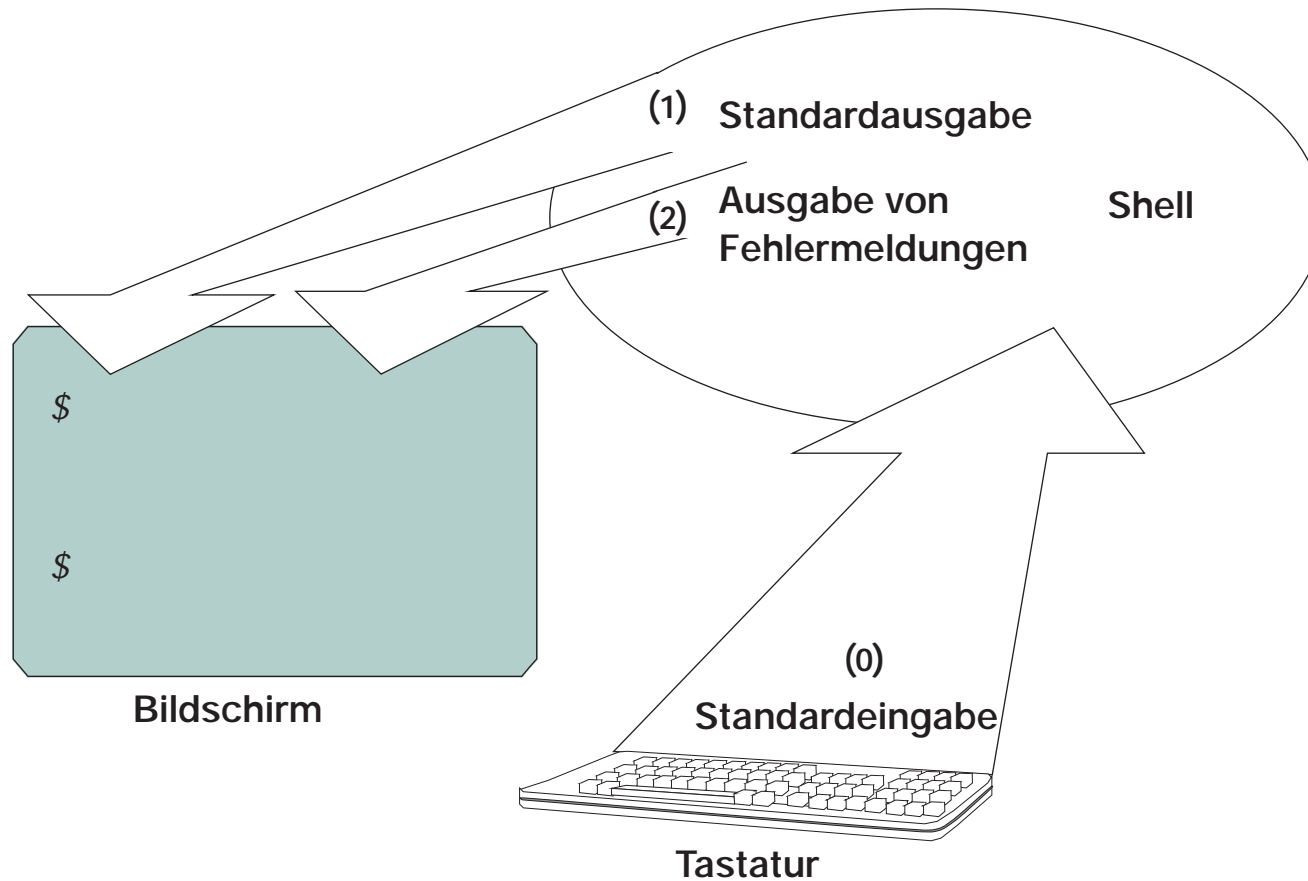
(ed, grep, sed, vi)

| Metazeichen | Suchbeispiel        | Bedeutung   |
|-------------|---------------------|---|
| .           | <code>/d/</code>    | Beliebiges einzelnes Zeichen<br>sucht nach allen Wörtern, die ein beliebiges Zeichen gefolgt von einem d, enthalten   |
| *           | <code>/m*/</code>   | Wiederholung des vorangestellten Zeichens<br>sucht nach einem oder hintereinander mehrfach auftretenden „m“<br>sucht nach Zeichenfolgen, in denen ein „m“ vorkommt<br>steht für mehrere beliebige Zeichen |
| .*          | <code>/.*/</code>   |   |
| ^           | <code>/^ */</code>  | Nachfolgender Suchbegriff steht am<br>Anfang einer Zeile  |
| \$          | <code>/DM\$/</code> | Nachfolgender Suchbegriff steht am Ende einer Zeile<br>sucht nach einer Zeile, die mit DM endet   |

## Metazeichen (Fortsetzung)

|              |  |  |
|--------------|--|--|
| <b>[ ]</b>   | <b>/[a-z]/</b><br><b>/[137]/</b>                   | Suchen eines in der Klammer angegebenen Zeichens<br>sucht nach einem beliebigen Kleinbuchstaben (z. B. a oder b)<br>sucht nach Ziffern 1, 3 oder 7 |
| <b>\</b>     | <b>/\*/</b>  | Fluchtsymbol - Aufheben einer evtl. Sonderfunktion des nachfolgenden Zeichens<br>sucht nach einem „*“  |
| <b>&amp;</b> | <b>s/Muster/&amp;.. /</b><br><b>s/Herr/&amp;n/</b> | Einsetzen des gefundenen Musters als neuen Begriff<br>ersetzt „Herr“ in „Herrn“  |

# 7 Bash Shell Eingabe/ Ausgabeumlenkung



# Ausgabeumleitung

*Kommando > Datei oder Gerätenamen*

**>**, um die Ausgabe eines Kommandos umzuleiten

```
$ ls > inhalt
```

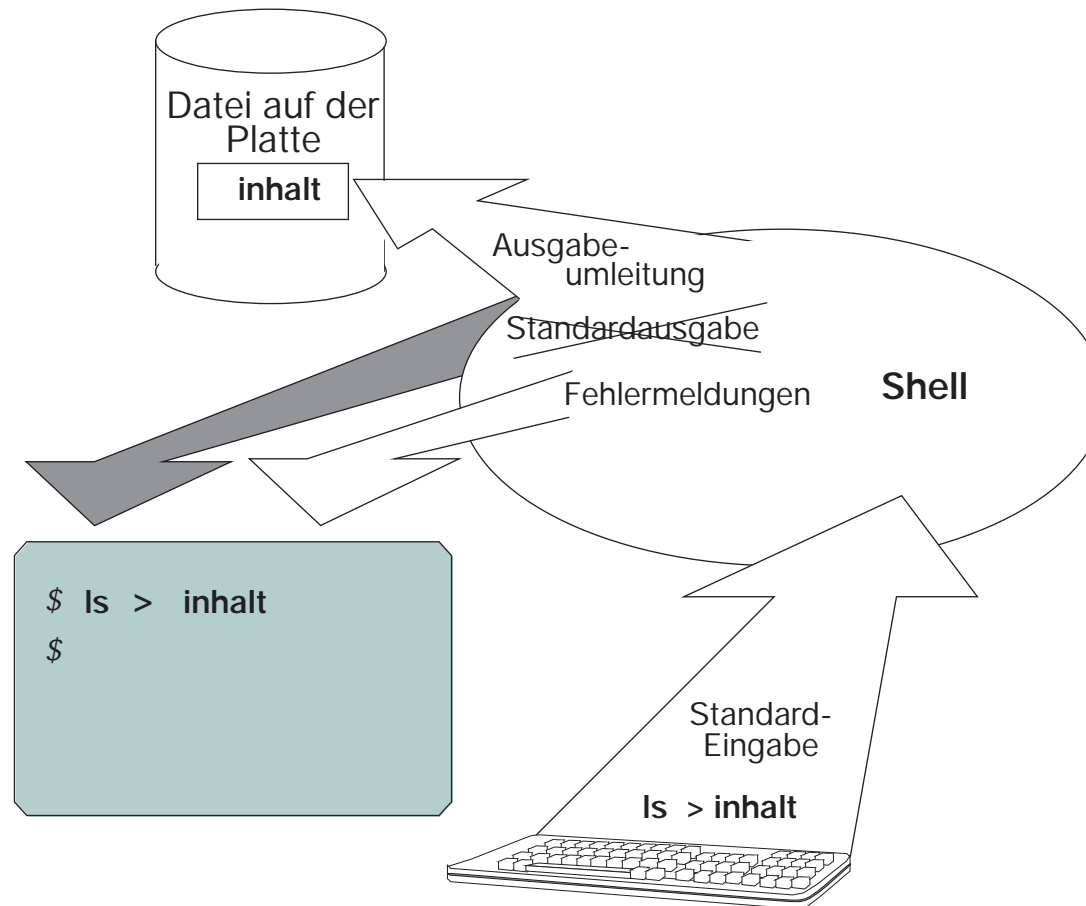
Umleitung der Ausgabe von ls

```
$ cat inhalt
```

```
befehle  
projektA  
ueb1  
$
```

Anzeige des Dateiinhalts von  
>inhalt<  
Die Dateinamen stehen hier un-  
tereinander, da nur die Ausgabe  
bei ls in Spalten aufgeteilt erfolgt

# Beispiel einer Ausgabe-Umlenkung



# Anhängen an eine Datei

*Kommando >> bestehende Datei*

```
$ cat inhalt >> neu
```

```
$ cat neu
```

*Alle Zeilen, die Sie eintippen  
werden in die Datei neu geschrieben.  
So auch dieser Text.*

*Um den Text abzuschliessen, geben  
Sie das Zeichen für Dateiende ein.  
Wie bei dem Kommando mail muss das  
Zeichen für Dateiende in einer eigenen  
Zeile stehen*

```
befehle  
projektA  
ueb1  
Wed Jun 11 17:10:00 MET 1991
```

```
$
```

Text der bisherigen  
Datei neu

angehängter  
Text aus der  
Datei inhalt

# Eingabeumlenkung

*Kommando < datei*

```
$ mail gisela < text1
```

```
$
```

# Fehlerumlenkung

*Kommando* **2>** *Dateiname oder Gerät*

```
$ cat inhalt dtum 2> fehler
befehle
projektA
ueb1
Wed Jun 11 17:10:00
```

Eine Datei dtum gibt es nicht

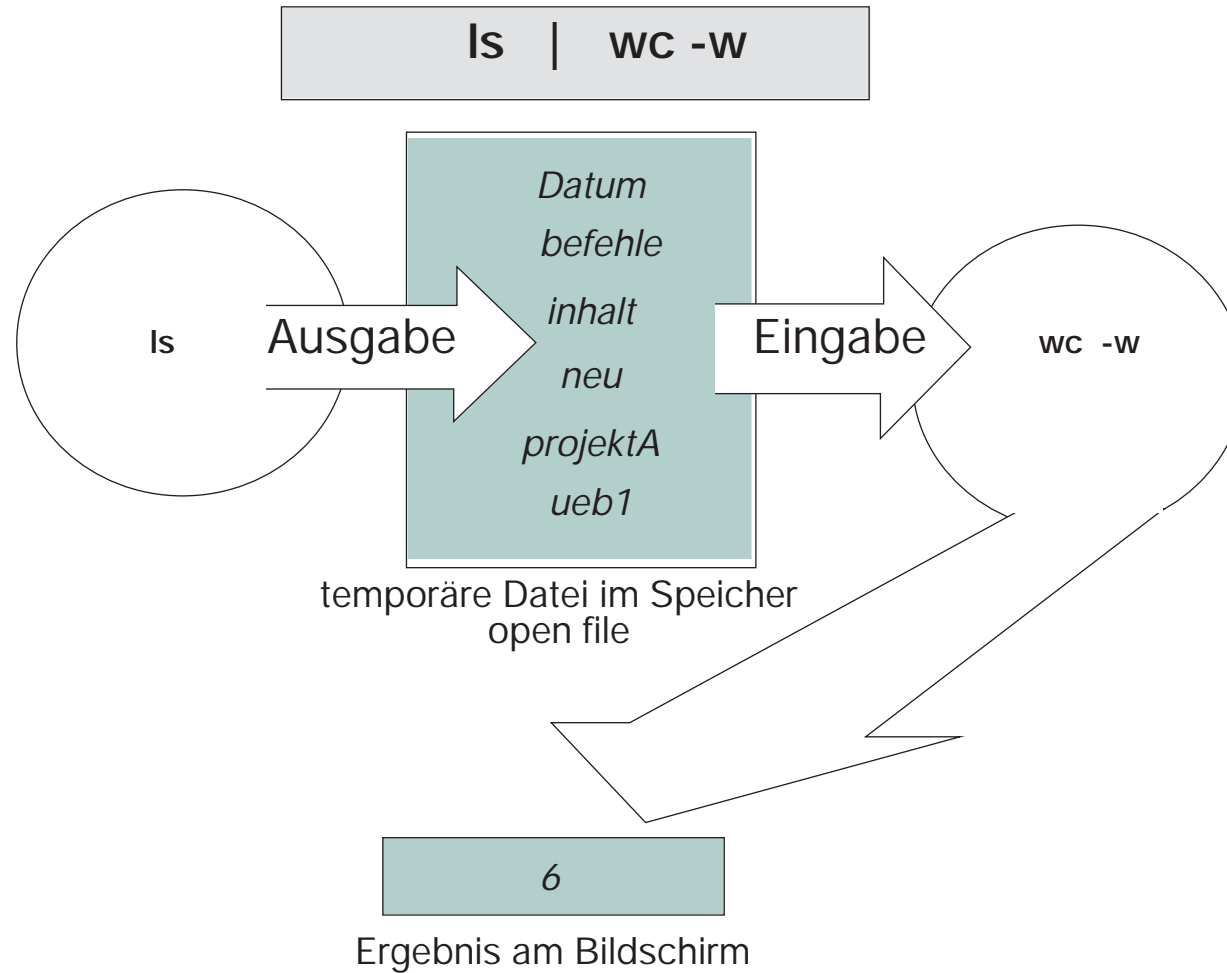
```
$ cat fehler
cat: cannot open dtum
```

Die Fehlermeldung erscheint nicht am Bildschirm, sondern wurde in die Datei *fehler* umgeleitet.  
Mit cat wird der Inhalt der Datei *fehler* angezeigt

*Kommando* **2>** **/dev/null**

Fehlermeldung "wegwerfen"

# Pipelining



# Filterkommandos

**sort** – Kommando, um Dateiinhalte zu sortieren

```
sort [-frn] [Dateiname(n)]
```

*n* **numeric string**

Am Zeilenanfang beginnende  
Zahlen, die durch Leerzeichen von anderen  
Zeichen getrennt sind, werden in arith-  
metischer Reihenfolge sortiert

**r** *reverse* – Es wird in umgekehrter Richtung sortiert

**f** *fold* – Klein- u. Großbuchstaben sind gleichberechtigt

# grep – Kommando

Nach Mustern in Dateien zu suchen

```
grep "Muster" [Dateiname(n)]  
get regular expression
```

hebt das Ende-Zeile-Zeichen auf

```
$ cat Datum inhalt neu \  
> ueb1 | grep "Zeile stehen"  
  
Zeile stehen  
  
$
```

Kennzeichen, daß die Shell  
weitere Eingaben für einen  
Kommandoaufruf erwartet (>)

Inhalt der Datei *neu*:

*Alle Zeilen, die Sie eintippen,  
werden in die Datei neu geschrieben.  
So auch dieser Text.  
Um den Text abzuschliessen, geben  
Sie das Zeichen für Dateiende ein.  
Wie bei dem Kommando mail muß das  
Zeichen für Dateiende in einer eigenen  
Zeile stehen*  
*befehle  
projektA  
ueb1  
Wed Jun 11 17:10:00 MET*

# 8 Shell - Metazeichen/Suchmuster

## Dateinamenexpansion

| Metazeichen | Ersetzung   |
|-------------|---|
| *           | steht für eine beliebige Zeichenfolge (oder kein Zeichen)<br>Allerdings nicht der Punkt zu Beginn einer Datei   |
| ?           | steht für <b>ein</b> einzelnes <b>beliebiges</b> Zeichen<br>(aber <b>nicht</b> für ein Leerzeichen)   |
| [ ... ]     | Die Klammer wird ersetzt durch <b>ein</b> in der Klammer angegebenes Zeichen. z.B.:<br><br>[abc]      wird ersetzt durch a oder b oder c<br>[1-3]      wird ersetzt durch 1 oder 2 oder 3   |
| [!... ]     | Die Klammer mit Ausrufezeichen wird ersetzt durch <b>ein</b> beliebiges Zeichen, das <b>nicht in der Klammer angegeben ist</b> . z.B.:<br><br>[!abc ]    wird ersetzt durch ein beliebiges Zeichen, aber nicht a oder b oder c<br>[! 1-3 ]    wird ersetzt durch ein beliebiges Zeichen, aber nicht 1 oder 2 oder 3 |

## Aufhebung eines Sonderzeichens (z.B. Metazeichen)

\

Dieser Schrägstrich (*backslash*) hebt den Ersetzungsmechanismus für das nachfolgende Metazeichen auf (*Fluchtsymbol*) z. B.:

\? Das Fragezeichen wird **nicht** durch ein beliebiges Zeichen ersetzt, sondern als Fragezeichen übernommen

## Weitere Ersetzungsmechanismen

*" text ..text "*

Übergabe von bestimmten Sonderzeichen,  
keine Ersetzung von Metazeichen: \* ? [ ]  
aber Ersetzung von Shellvariablen \$ und  
**Kommandosubstitution** ` \ bzw. \$( )

*' text ..text '*

keinerlei Substitution  
Metazeichen, Variablen, Ergebnisse von  
Kommandos werden **nicht** ersetzt

# Beispiele

Hebt das Ende-Zeile-Zeichen für die Shell auf

```
$ echo * welche Nachricht erhalten \  
> Sie ?
```

Ersetzung des \* durch alle Dateinamen vom aktuellen Directory

```
Datum befehle inhalt neu projektAueb1  
welche Nachricht erhalten Sie ?
```

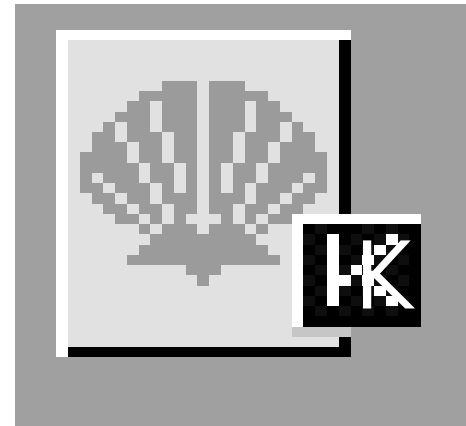
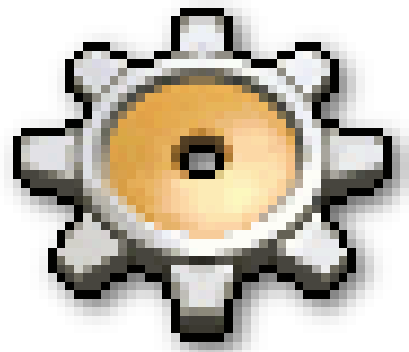
Kennzeichen, daß die Shell weitere Eingaben für einen Kommandoaufruf erwartet (>)

Wären 1-stellige Dateinamen im aktuellen Directory vorhanden, würde auch das Fragezeichen durch diese Dateinamen ersetzt werden.

```
$ echo " * welche Nachricht erhalten \  
> Sie ? "  
* welche Nachricht erhalten  
Sie ?
```

# 9 Erleichterungen mit der Bash

(bzw. der Korn-Shell)




## Übersicht einiger Shells

| Name der Shell                             | Aufruf Prompt               | Kurze Beschreibung  | Charakteristisch                      |
|--|-----------------------------|---|---------------------------------------|
| <b>Bourne-Shell</b>                        | <code>/bin/sh</code><br>\$  | ursprüngliche Shell<br>auf allen Rechnern verfügbar<br>Grundlage für<br>systemspezifische Abläufe   | nicht sehr<br>benutzerfreundlich      |
| <b>Korn-Shell</b>                          | <code>/bin/ksh</code><br>\$ | kompatibel zur Bourne-Shell<br>mit wesentlichen Ergänzungen u. a. Historie-<br>Mechanismus, Alias, Befehlszeileneditor                              | komfortabel und mächtig               |
| <b>Bash</b><br><i>(bourne again shell)</i> | <code>/bin/sh</code><br>\$  | meist unter LINUX eingesetzt<br>kompatibel zur Bourne-Shell noch mehr<br>Funktionen wie die Korn-Shell  | noch komfortabler                     |
| <b>C-Shell</b>                             | <code>/bin/csh</code><br>%  | Ähnlichkeit zur Programmier-sprache C –<br>ebenso Historie-Mechanismus, Alias,<br>Befehlszeileneditor – aber nicht kompatibel<br>zur Bourne-Shell ! | wird gerne von Entwicklern<br>genutzt |
| <b>TC-Shell</b>                            | <code>/bin/tcsh</code><br>% | erweiterte C-Shell - mehr Komfort<br>nicht kompatibel zur Bourne-Shell !  | wird gerne von Entwicklern<br>genutzt |

## Vorteile der bash (bzw. ksh)

- ✦ kompatibel zur Bourne-Shell (if, while, for, case)
- ✦ Kommandohistorie  
Wiederholung von Kommandos
- ✦ Editieren der Kommandozeilen über vi oder emacs
- ✦ Alias-Mechanismus
- ✦ Weitere Namensexpandierung  
z.B. ~benutzer für \$HOME des angegebenen Benutzers
- ✦ Umfangreiche Voreinstellungen über Variable und Optionen  
**z.B.** Schutz vor Überschreiben von bestehenden Dateien durch Setzen von  
set -o noclobber  
z.B. Prompt mit Pfadangabe  
PS1='\$PWD'
- ✦ Erweiterte Programmkontrolle Stoppen, Vorder-/Hintergrund

## Starten und Abmelden einer bash (bzw ksh)

- ❑ Entweder als Login-Shell  
(Eintrag unter /etc/passwd)  
Beim Anmelden wird **/etc/profile** und **\$HOME/.profile** und **\$HOME/.kshrc** bzw. bei bash: **\$HOME/.bashrc** gelesen  
(nur bei Korn-Shell: falls der Wert der Variablen ENV zugewiesen wurde  
(ENV=\$HOME/.kshrc; export ENV))
- ❑ oder als Sub-Shell mit dem Aufruf ksh bzw. bash (unter SuSE sh)
- ❑ Bei jedem Start einer ksh oder bash (z.B. neues Fenster oder bei Ausführung einer Kommandoprozedur) wird die Datei \$HOME/.kshrc bzw. \$HOME/.bashrc gelesen  
(nur bei Korn-Shell: wenn die Variable ENV entsprechend gesetzt wurde)
- ❑ Das Promptzeichen der ksh ist: \$, (bei bash (SuSE) bereits voreingestellt z. B. „Name@rechner.>“)  
bzw. durch Voreinstellung in **.kshrc (.bashrc)** kann als Prompt z.B. der **aktuelle Pfadname** angezeigt werden 
- ❑ Die ksh oder bash wird durch **exit** oder <Ctrl +d> beendet

# Der Historie-Mechanismus



- ❑ Alle aufgerufenen Kommandos in `$HOME/.bash_history` (`.sh_history`)
- ❑ Die Variable `HISTSIZE`  
(Anzahl der Kommandos für späteren Zugriff z.B. `HISTSIZE=20`)
- ❑ Die zuletzt eingegebenen Kommandos können nochmals abgerufen werden
  - mit der laufenden Nummer: `r nummer`
  - oder relativ z.B. das drittletzte Kommando: `r -3`
  - oder das letzte Kommando `r`
  - oder ein Kommando mit Namensbeginn `r name`
  - Befehl um die Datei `.sh_history` zu bearbeiten  
lautet: `fc`, `r` ist ein vorgegebener alias auf `fc -e`
- ❑ `history -10` oder `fc -l -10` zeigt die letzten 10 Befehle an  
`history 6 8` oder `fc -l 6 8` zeigt den 6. und 8. Befehl an

# Unter Bash einfache Korrekturen und Ergänzungen

| Funktion   | Tasten                     |
|--|----------------------------|
| In zuletzt eingegebenen Kommandos blättern   | ↑ ↓                        |
| Innerhalb der Kommandozeile vor und zurück   | ← →                        |
| Cursor an den Beginn/das Ende der Zeile<br>oder  | <Ctrl+a>, <Ctrl+e>         |
| Wortweise zurück ( <i>backwards</i> )<br>bzw. vorwärts ( <i>forward</i> )  | <Alt+b>, <Alt+f>           |
| Zeichen vorwärts/rückwärts löschen   | BkSp, Ent                  |
| Wort löschen   | <Alt+d>                    |
| Bis Ende der Zeile löschen   | <Ctrl+k>                   |
| Zeichen tauschen   | <Ctrl+t>                   |
| Wörter vertauschen   | <Alt+t>                    |
| Suche nach früheren Kommandos (Angabe des Buchstabens) Weitersuchen: nochmalige Eingabe von Ctrl+r<br>Wenn gefunden, Kommando mit Tab aktivieren | <Ctrl+r><br><br><b>TAB</b> |

## Kommandozeilen in der ksh editieren?

(Unter der Bash bereits voreingestellt über Cursorstasten - siehe vorige Seite)

- ❑ Voraussetzung: Variable setzen

**export VISUAL=vi** oder **export VISUAL=emacs** oder **set -o vi**

- ❑ Zum Editieren Escape-Taste drücken, dann sind folgende Funktionen bei Visual=vi möglich:

| Funktion - Positionieren           | Taste |
|------------------------------------|-------|
| Vorherige Befehlszeile (nach oben) | k     |
| Nächste Befehlszeile (nach unten)  | j     |
| Zeichen nach rechts wandern        | l     |
| Zeichen nach links wandern         | h     |
| Anfang der Zeile                   | ^     |
| Ende der Zeile                     | \$    |

## Weitere vi-Funktionen

| <b>Funktion - Positionieren</b> | <b>Taste</b> |
|---------------------------------|--------------|
| 1 Wort nach links wandern       | <b>b</b>     |
| 1 Wort nach rechts wandern      | <b>w</b>     |
| nach Zeichen z.B. x suchen      | <b>fx</b>    |
| <b>Funktion - Korrigieren</b>   |              |
| Zeichen löschen                 | <b>x</b>     |
| Wort löschen                    | <b>dw</b>    |
| bis zum Ende der Zeile löschen  | <b>d\$</b>   |
| zuletzt Gelöschtes einfügen     | <b>p</b>     |

# Expandierung von Pfadnamen innerhalb des Befehlszeileneditors

(ESC-Taste) und:

| Befehl | Wirkung                        |
|--------|--------------------------------|
| =      | Auflisten von Pfadnamen        |
| \      | Vervollständigen von Pfadnamen |
| *      | Expandierung von Pfadnamen     |

## Tilde-Expandierung (für bash und ksh)

| Kürzel                   | Wird ersetzt durch<br>vollständigen Pfadnamen von     |
|--------------------------|---|
| ~                        | Home-Directory des Benutzers                          |
| ~+                       | aktuelles Arbeitsverzeichnis                          |
| ~-                       | vorheriges Arbeitsverzeichnis<br>(vor dem letzten cd) |
| <b>~<i>loginname</i></b> | Home-Directory des angegebenen<br>Benutzers           |

□ zusätzlich mit

**cd -**

**in vorhergehendes Verzeichnis**

# Die Alias-Funktion

Kurzform für Kommandos  
bei häufig benutzen Kommandos zu empfehlen:

**alias** *kürzel='Kommandofolge'*

**z.B.**

```
$ alias rm='rm -i'
```

```
$ alias ll='ls -la'
```

# Alias-Befehle

Auch hier unter der Bash bereits einige Aliase voreingestellt (z.B. ll)

| Befehl                              | Beispiel  | Auswirkung   |
|-------------------------------------|---|--|
| <b>alias <i>Kürzel</i>="Befehl"</b> | <b>alias ll="ls -l"</b><br><b>alias rm="rm -i"</b>                                  | Bildung eines Alias                                  |
| <b>\Kürzel</b>                      | <b>\rm <i>datei</i></b>   | Aufhebung der Aliasfunktion für den aktuellen Befehl |
| <b>unalias <i>Kürzel</i></b>        | <b>unalias rm</b>   | Aliasfunktion aufheben                               |
| <b>alias -x <i>Kürzel</i></b>       | <b>alias -x ll="ls -l"</b>  | Aliasfunktion auch für Subshells exportieren         |
| <b>alias</b>                        | <b>alias</b><br><i>integer=typeset -i</i><br><i>history=fc -l</i><br><i>r=fc -e</i> | Anzeige der vorhandenen Aliasfunktionen              |

## Zusätzliche Optionen

- ❑ Mit dem Kommando **set -o** können folgende z.B. Optionen gesetzt werden:

| Option                      | Bedeutung   |
|-----------------------------|---|
| <b>allexport</b>            | Alle gebildeten Variablen werden grundsätzlich exportiert                                     |
| <b>bgnice</b>               | Hintergrundprozesse laufen mit einer niedrigeren Priorität                                    |
| <b>emacs</b> <i>oder vi</i> | Der angegebene Editor wird zum Editieren der Befehlszeile genommen (Korrektur über ESC-Taste) |
| <b>ignoreeof</b>            | Mit der Tastenkombination Ctrl + d kann die Shell nicht mehr beendet werden (wird ignoriert)  |
| <b>noclobber</b>            | Bereits bestehende Dateien können über Umleitungszeichen > nicht mehr überschrieben werden    |

- ❑ mit **set +o** können gesetzte Optionen wieder ausgeschaltet werden

# Verwendung von Optionen

- ❑ Eingabe über Terminal - nur für die aktuelle Sitzung
- ❑ Eintrag in die Datei \$HOME/.kshrc  
z.B.

```
set -o allexport  
set +o ignoreeof  
set -o vi
```

# 10 Variablen- und Kommandosubstitution

## Was sind Variable?

- ❑ veränderliche Werte      z.B. Zuweisung von Namen (Schreibersparnis) und Zuweisung von Werten innerhalb Shell-Scripts mit read (Lesen der Antwort)  
Beispiel: k=/home/kurs, ...; read Antwort;
- ❑ Systemvariable      vordefinierte Variable  
z.B. Home-Directory, Pfadname für Befehlssuche  
\$HOME, \$PATH, \$USER ...  
Abfrage durch Programme deshalb Vorsicht bei Änderung)
- ❑ Positionsparameter      für Abarbeitung der einem Befehl mitgegebenen Werte (Parameter) (Shell-Scripts)  
\$1, \$2, ... \$\*

# Wie werden Variable zugeordnet?

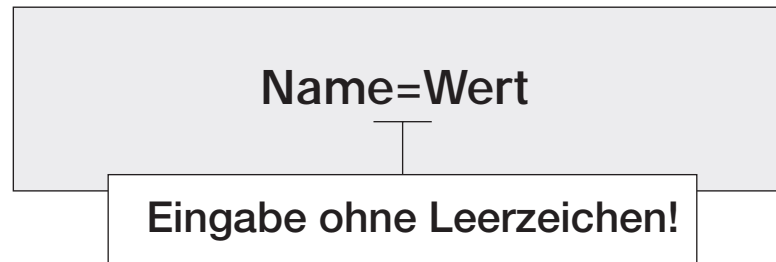
1. ***name=wert***  
nur für die lokale Shell-Umgebung
2. **export name=wert** oder unter linux **declare**  
auch für Subshells/Unterprogramme
3. über typeset **Option Variable[=*wert*]**

| Option    | Ableitung von                  | Auswirkung auf die Variable                |
|-----------|--------------------------------|--|
| <b>-i</b> | <i>integer</i>                 | nur ganze Zahlen werden akzeptiert         |
| <b>-u</b> | <i>uppercase</i> (nicht linux) | Eingabe wird in Großbuchstaben zugewiesen  |
| <b>-l</b> | <i>lowercase</i> (nicht linux) | Eingabe wird in Kleinbuchstaben zugewiesen |
| <b>-x</b> | <i>export</i>                  | Variable wird exportiert                   |

4. **innerhalb von Shell-Prozeduren zusätzlich**
  - mit dem Kommando `read Variable`
  - automatisch über Parameter (Position 1 entspricht \$1 usw.)

# Wie arbeiten Sie mit Variablen?

- ❑ Zuweisung z.B.



## ❑ Ersetzung der Variablen

**\$Name**

*oder um evtl. Ergänzungen anzuhängen*

**\${Name}**

## Beispiel 1

```
$ k=/home/kurs
$ echo "hallo $k"
hallo /home/kurs
$ pwd
/home/kurs/monika
$ cd $k/hans
$ pwd
/home/kurs/hans
```

Variablen werden auch innerhalb von  
Anführungszeichen " " ersetzt!

## Beispiel 2

```
$ b=Benutzer  
$ echo Hallo ${b}in  
Hallo Benutzerin
```

**echo**      [ *Text ... Text* ]

*Widerhall, Kopie*

alle Wörter des Textes werden von  
der Shell als Nachricht wiedergegeben

In der Texteingabe bedeuten folgende Steuerzeichen:

|                   |                                 |
|-------------------|---------------------------------|
| <code>\007</code> | Klingelzeichen                  |
| <code>\c</code>   | Cursor bleibt in gleicher Zeile |
| <code>\n</code>   | neue Zeile                      |
| <code>\t</code>   | Tabulator                       |

## Einige vordefinierte Variable

|                  |   |
|------------------|---|
| <b>\$HOME</b>    | absolute Pfadname des Login-Directories aus der <i>/etc/passwd</i>  |
| <b>\$IFS</b>     | Hier sind die Separatorzeichen zugewiesen<br>Standard: Leerzeichen, Tabulator und Neue Zeile  |
| <b>\$LANG</b>    | Soweit im System enthalten - Zuordnung der jeweiligen Sprache   |
| <b>\$LOGNAME</b> | Login-Name des Benutzers  |
| <b>\$PATH</b>    | Pfadnamen, unter denen die Shell nach Kommandos sucht<br><b>PATH=/usr/bin:/bin:/usr/ucb/bin</b><br><i>Beispiel einer Ergänzung:</i><br><b>PATH=\$PATH:\$HOME/Eigenbefehle</b> |
| <b>\$PS1</b>     | Bereitzeichen der Shell<br><b>PS1='\$PWD &gt; '</b> nur ksh<br>z. B. <i>/home/kurs/hans/Texte &gt;</i>  |
| <b>\$TERM</b>    | Terminaltyp der Dialogstation<br>wichtig bei vielen bildschirmorientierten Programmen<br>(z. B. grafische Oberfläche, vi, more etc.)  |

## Spezielle vordefinierte Variable

set

env

| Variable        | Bedeutung   |
|-----------------|---|
| <b>\$ENV</b>    | Name der Datei zugeordnet, der ksh-spezifische Voreinstellung enthält, z.B. \$HOME/.kshrc<br>in .profile gesetzt setzen |
| <b>\$VISUAL</b> | vi oder emacs<br>für Befehlszeilen-Editor   |
| <b>\$PWD</b>    | das aktuelle Directory  |
| <b>\$OLDPWD</b> | das vor einem cd benutzte Directory   |

**set** – Kommando, um definierte Variable der aktuellen Shell anzuzeigen

**env** – Kommando, um definierte und exportierte Variablen anzuzeigen

## Wie kann das Promptzeichen verändert werden?

- ❑ Über Zuweisung der Variablen PS1  
z.B. in .kshrc

```
export PS1=' $PWD > '
```

z.B. `/home/kurs/ben01>`

- ❑ Über Zeichenkettenmanipulation einer Variablen (##) kann der Pfad so abgeändert werden, daß nur das Subdirectory angezeigt wird:

```
export PS1='.../${PWD##*/} > '
```

z.B. `.../ben01>`

- ❑ Für Bash sie auch man bash (Suchen nach Prompting)

## Löschen von gesetzten Variablen

`unset Variable`

# Kommandosubstitution

|                         |   |
|-------------------------|---|
| <code>\Kommando\</code> | Ersetzung durch das Ergebnis<br>des angegebenen Kommandos |
|-------------------------|---|

oder in einer neueren Schreibweise:

```
$( Kommando )
```



Unbedingt auf die Leerzeichen nach und vor der Klammerung achten!

# Beispiel Kommandosubstitution

Das Ergebnis des Kommandos ›date‹ soll hier eingesetzt werden.

```
$ echo "hallo $k heute am `date` \  
> arbeiten `who | wc -l` Benutzer "  
  
hallo /home/kurs heute am Fri Jun 20 16:47:53  
MET 1996 arbeiten      4 Benutzer  
  
$
```

der Backslash › \  
hebt das Zeilen-  
ende auf

Variable und Ergebnisse von Kommandos werden innerhalb von doppelten Anführungszeichen "..." ersetzt!

# Ersetzungsmechanismen

|                              |  |
|------------------------------|--|
| <code>\</code>               | Aufhebung des nachfolgenden Sonderzeichens/<br>Sonderbedeutung (z. B. Alias)<br>(z. B. auch Ende einer Zeile)  |
| <code>"text ... text"</code> | Keine Ersetzung der Metazeichen<br>(Dateinamenexpansion): * ? [ ]<br>Aber :<br>Ersetzung von Shell-Variablen (\$)<br>Ersetzung durch Ergebnisse von Kommandos<br><code>`</code> , bzw. <code>\$( )</code><br>Ersetzung von Alias |
| <code>\$name</code>          | Ersetzt wird der zugewiesene Wert der Variablen <i>"name"</i>  |
| <code>\$( Kommando )</code>  | Ersetzt wird das Ergebnis des angegebenen Kommandos  |
| <code>'Kommando'</code>      | gleich wie oben  |
| <code>`text ... text`</code> | Keinerlei Substitution   |

# Voreinstellung für Bash und Korn-Shell

Für alle Shells verfügbare Aliase und spezielle set-Optionen

- ❑ Im .profile die Variable ENV=~/.kshrc (nur für Korn-Shell)
- ❑ Alias-Einträge in ~/.kshrc bzw. für die Bash in ~/.bashrc

**Beispiel einer .kshrc**

```
$ cat .kshrc  
  
alias ll='ls -lFa'  
set -o vi  
export PS1= '$PWD > '
```

# Der Datentyp integer

- ❑ Es werden nur Zahlen akzeptiert
- ❑ Mit Integer-Variablen kann bei Zuweisung von Variablen direkt gerechnet werden (also ohne Kommando let)

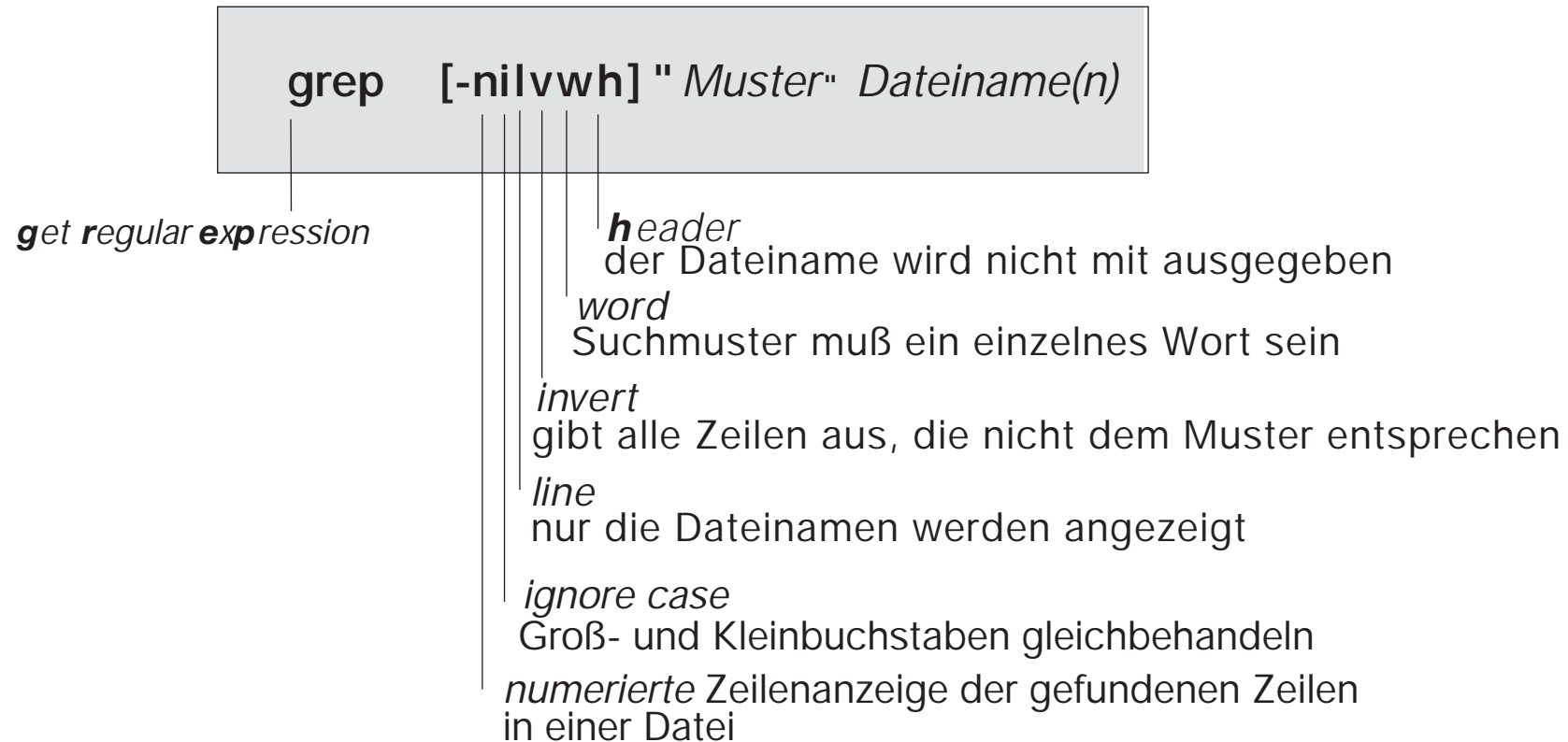
integer ist unter Korn-Shell die Aliasbezeichnung von typeset -i  
unter bash wird meist statt typeset **declare -i** verwendet

| Opera-<br>tor | Auswirkung     | Beispiel           | Ergebnis<br>echo \$erg |
|---------------|----------------|--------------------|------------------------|
| +             | Addition       | typeset -i erg=5+3 | 8                      |
| -             | Subtraktion    | erg=erg-3          | 5                      |
| *             | Multiplikation | erg=erg*3          | 15                     |
| /             | Division       | typeset -i erg=5/3 | 1                      |
| %             | Modulo         | typeset -i erg=5%3 | 2                      |

Das \$-Zeichen für Variablen braucht bei der Berechnung nicht angegeben zu werden

# 11 Werkzeuge und Dienstprogramme

## Suchen von Zeilen innerhalb einer Datei



# Metazeichen – reguläre Ausdrücke

(ed, grep, sed, vi)

| Metazeichen | Suchbeispiel       | Bedeutung   |
|-------------|--------------------|---|
| .           | <code>/d/</code>   | Beliebiges einzelnes Zeichen<br>sucht nach allen Wörtern, die ein beliebiges Zeichen gefolgt von einem d, enthalten |
| *           | <code>/m*/</code>  | Wiederholung des vorangestellten Zeichens<br>sucht nach einem oder hintereinander mehrfach auftretenden „m“         |
| .*          | <code>/.*/</code>  | sucht nach Zeichenfolgen, in denen ein „m“ vorkommt<br>steht für mehrere beliebige Zeichen                          |
| ^           | <code>/^ */</code> | Nachfolgender Suchbegriff steht am Anfang einer Zeile   |

## Metazeichen (Fortsetzung)

| Metazeichen | Suchbeispiel  | Bedeutung  |
|-------------|---|--|
| \$          | <code>/DM\$/</code>   | Nachfolgender Suchbegriff steht am Ende einer Zeile<br>sucht nach einer Zeile, die mit DM endet  |
| [ ]         | <code>/[a-z]/</code><br><code>/[137]/</code>                  | Suchen eines in der Klammer angegebenen Zeichens<br>sucht nach einem beliebigen Kleinbuchstaben (z. B. a oder b)<br>sucht nach Ziffern 1, 3 oder 7 |
| \           | <code>/\*/</code>   | Fluchtsymbol - Aufheben einer evtl. Sonderfunktion des nachfolgenden Zeichens<br>sucht nach einem „*“  |
| &           | <code>s/Muster/&amp;../</code><br><code>s/Herr/&amp;n/</code> | Einsetzen des gefundenen Musters als neuen Begriff<br>ersetzt „Herr“ in „Herrn“  |

## Beispiel

```
$ grep "^ben0[1-3]" /etc/passwd ...  
ben01:xcziui:101:20:Meier, Eva, Tel.446:/home/kurs/ben01:/bin/ksh  
ben02:xcber:102:20:Huber, Hans, Tel. 143:/home/kurs/ben02:/bin/sh  
ben03:nmx fz:103:20:Beck, Ute, Tel. 222:/home/kurs/ben03:/bin/ksh  
$
```

# find – Kommando, um Dateien zu suchen und zu finden



relativer oder absoluter Pfadname des Directories, von dem ab gesucht werden soll, auch für das aktuelle Directory muß dann der Punkt (.) angegeben werden

**-name** *Dateiname*

**-user** *Benutzername*

**-mtime** *n (Anzahl der Tage )*

**-newer** *Dateiname*

Ohne Angabe von Suchkriterien werden alle Dateien gesucht und mit der Ausgabeart >-print< angezeigt.

**-print**

Anzeige der gefundenen Dateien

**-exec** *Befehl { } ";"*  
(ausführen)

Ausführung der nachfolgenden Befehle

## Beispiel von find

```
$ find /home/kurs -name sicher -print
/home/kurs/monika/Befehle/sicher
$ cd
$ pwd
/home/kurs/monika
$ find . -name "s*" -print
./Texte/sprueche
./Befehle/sicher
$ find . -print
./ProjektA
./ProjektA/beispiel.c
./ProjektA/io.c
./ProjektA/pro.c
./Texte
./Texte/kekse
./Texte/sprueche
./Uebungen
./Uebungen/Datum
./Uebungen/inhalt
./Uebungen/neu
./Uebungen/neuer
./Uebungen/ueb1
./Uebungen/datum
./Befehle
./Befehle/sicher
```

Angabe des gefundenen Dateinamens mit dem absoluten Pfadnamen /home/kurs...

Suche nach allen Namen mit »s« beginnend  
Angabe der gefundenen Dateinamen mit dem relativen Pfadnamen ab Suchposition ./

Anzeige aller Dateien mit dem Pfadnamen ab Suchposition, die rekursiv in dem angegebenen Directory gefunden wurden.

## find mit grep kombiniert

```
$ find /home/kurs -user monika -exec grep "denkst$" {} ";"  
/home/kurs/monika/Texte/sprueche: Alles dauert länger als Du denkst
```

Zeigt alle Zeilen von Dateien ab Directory /home/kurs, die monika gehören und bei denen am Zeilenende denkst steht

# Ausschneiden von Spalten – cut

```
cut [-d Zeichen] fFeldnr. [ ,Feldnr., Feldnr, ...] [ Feldnr.-Feldnr]
```

*f*field

Spalten ( Felder) (und/oder ›von bis‹ z. B. 2,4,6-9)

*d*delimiter - Trennungszeichen

Achtung! Als Standardeinstellung Tabulatorzeichen,  
nicht wie bei sort Leerzeichen!

Kennzeichen für Trennungszeichen unterschiedlich!

## Beispiel

Aus der Datei /etc/passwd sollen nur die Spalten 1, 5 und 6 angezeigt werden:

| 1      | 2       | 3    | 4   | 5                      |
|--------|---------|------|-----|------------------------|
| ben01: | xcziui: | 101: | 20: | Meier, Eva, Tel.446:   |
| ben02: | xcber:  | 102: | 20: | Huber, Hans, Tel. 143: |
| ben03: | nmx fz: | 103: | 20: | Beck, Ute, Tel. 222:   |

← Spalten 1 - 5

```
ben01:/home/kurs/ben01:/bin/ksh
ben02:/home/kurs/ben02:/bin/sh
ben03:/home/kurs/ben03:/bin/ksh
```

Vorsicht: Bei manchen Systemen beginnt die Spaltenzählung bei cut mit **0**!

```
$ cut -d: -f1,5-6 /etc/passwd
```

```
ben01::Meier, Eva, Tel.446:/home/kurs/ben01:
ben02:Huber, Hans, Tel. 143:/home/kurs/
ben02:ben03:Beck, Ute, Tel. 222:/home/kurs/ben03:
$
```

# Das Sort-Programm

**sort [-frnb] -tZeichen -kPosition, Position] Datei [Dateiname(n)]**  
**+Position -Anzahl der Zeichen**

**Kriterien für den Sort** (der Buchstabe entfällt bei neueren Systemen - hier wird Spalte mit +Spaltennr. angegeben  
Von Position Zahl1.Zahl2 bis Position Zahl1.Zahl2  
Zahl1=SpaltenNr (beginnend mit 0)  
Zahl2=ZeichenNr innerhalb der Spalte (beginnend mit 0)

**Trennzeichen** (*termination character*)

Trennzeichen wenn nicht Leerzeichen als Separator dienen

**blank** (*Leerzeichen*) - führende Leerzeichen werden nicht mitsortiert

**numeric string**

Am Zeilenanfang beginnende Zahlen, die durch Leerzeichen von anderen Zeichen getrennt sind, werden in **arithmetischer** Reihenfolge sortiert

**reverse** - Es wird in umgekehrter Richtung sortiert

**fold** - Groß- und Kleinbuchstaben werden gleich behandelt

## Beispiel sort als Filterprogramm (Sortierung nach Spalten)

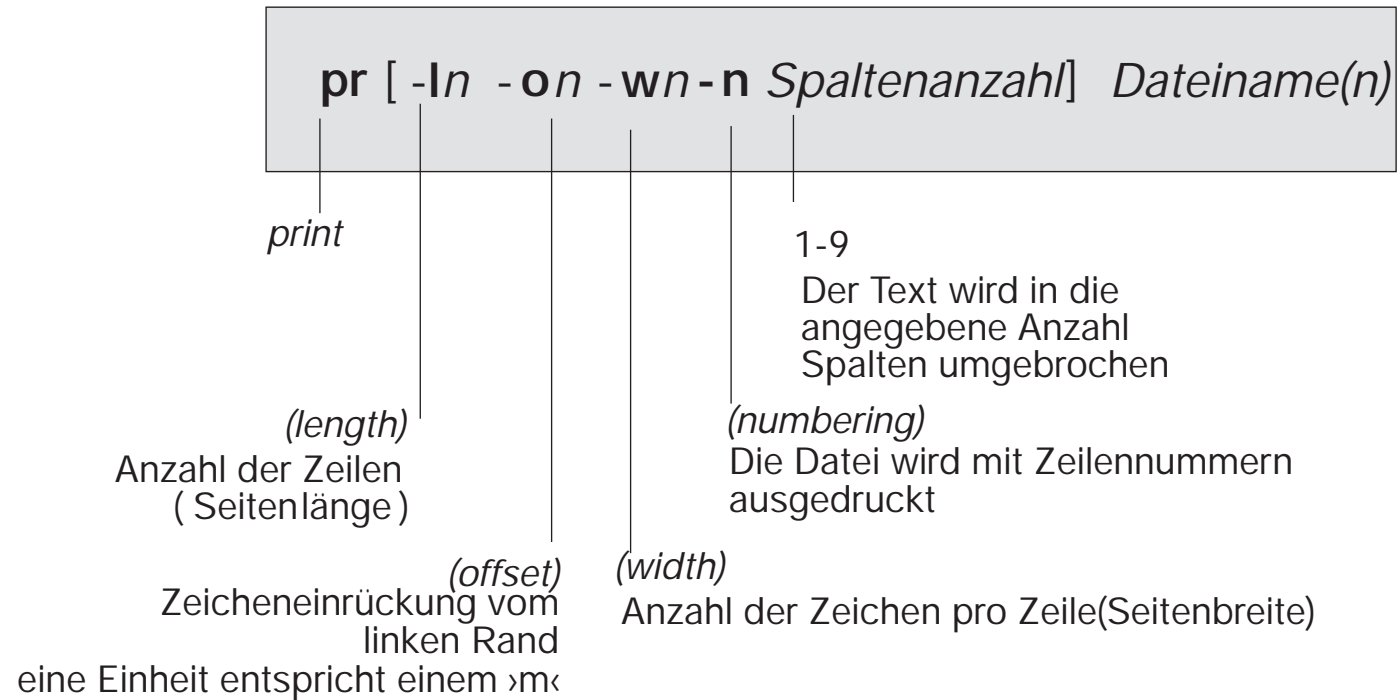
| 0     | 1       | 2    | 3   | 4                     |
|-------|---------|------|-----|-----------------------|
| ben01 | :xcziui | :101 | :20 | Meier, Eva, Tel.446   |
| ben02 | :xcber  | :102 | :20 | Huber, Hans, Tel. 143 |
| ben03 | :nmx fz | :103 | :20 | Beck, Ute, Tel. 222   |

← Spalten 0 - 4  
4.1 4.4 ← Position

ben01:/home/kurs/ben01:/bin/ksh  
ben02:/home/kurs/ben02:/bin/sh  
ben03:/home/kurs/ben03:/bin/ksh

```
$ grep "ben" /etc/passwd | sort -t: -k4.1,4.4 | tee bensort      ältere Version
$ grep "ben" /etc/passwd | sort -t: +4 -4 | tee bensort
...
ben03:nmx fz:103:20:Beck, Ute, Tel. 222:/home/kurs/ben03:/bin/ksh
ben02:xcber:102:20:Huber, Hans, Tel. 143:/home/kurs/ben02:/bin/sh
ben01:xcziui:101:20:Meier, Eva, Tel.446:/home/kurs/ben01:/bin/ksh
$
```

# Formatierte Ausgabe



# Ansehen von Dateiinhalten

**\$ more text1**

*Herzlich Willkommen zum UNIX-Einfuehrungskurs*

...  
**h**

-----

|                            |                                      |
|----------------------------|--------------------------------------|
| <i>h</i>                   | <i>help</i>                          |
| <i>q or Q</i>              | <i>quit</i>                          |
| <i>&lt;blank&gt; or \n</i> | <i>next page</i>                     |
| <i>l</i>                   | <i>next line</i>                     |
| <i>d or ^D</i>             | <i>display half a page more</i>      |
| <i>. or ^L</i>             | <i>redisplay current page</i>        |
| <i>f</i>                   | <i>skip the next page forward</i>    |
| <i>n</i>                   | <i>next file</i>                     |
| <i>p</i>                   | <i>previous file</i>                 |
| <i>\$</i>                  | <i>last page</i>                     |
| <i>s savelist</i>          | <i>save current file in savelist</i> |
| <i>/pattern/</i>           | <i>search forward for pattern</i>    |
| <i>?pattern? or</i>        | <i>search backward for pattern</i>   |
| <i>^pattern^</i>           |                                      |
| <i>!command</i>            | <i>execute command</i>               |
| <i>number</i>              | <i>linenumber</i>                    |

-----

*Hilfsfunktion*

*Beenden des Programms*

*nächste Seite (Leertaste/Return)*

*nächste Zeile*

*nächste halbe Seite*

*aktuelle Seite nochmal*

*überspringe nächste Seite*

*nächste Datei*

*vorhergehende Datei*

*letzte Seite*

*sichere aktuelle Datei in savelist*

*suche nach dem Muster*

*vorwärts*

*suche nach dem Muster*

*rückwärts*

*führe das Kommando aus*

*gehe nach Zeile n (n für Zahl)*

## Die wichtigsten Funktionen, die Sie benötigen werden, sind:

Leertaste

oder    Seitenweise vorwärts blättern

Returntaste

\$

Anzeige der letzten Seite (*Ende des Textes*)

1

Zurück zur Zeile 1

n

Gehe zur Zeile n (*z. B. für n=21: zur Zeile 21*)

q

Beenden des Programmes (*quit*)

*alternativ zu less das Kommando **more** (meist in **man** integriert –  
bei more Leertaste um neue Seite zu sehen!)*

## Nur ein kleiner Blick in die Dateien:

`head [- n] datei(en)`

*Kopf*

*number*

Anzahl der ersten Zeilen

ohne Angabe werden die ersten 10 Zeilen gezeigt

um die ersten Zeilen einer Datei anzusehen

`tail [-f] [- n] datei(en)`

*Schwanz*

*number*

Anzahl der letzten Zeilen

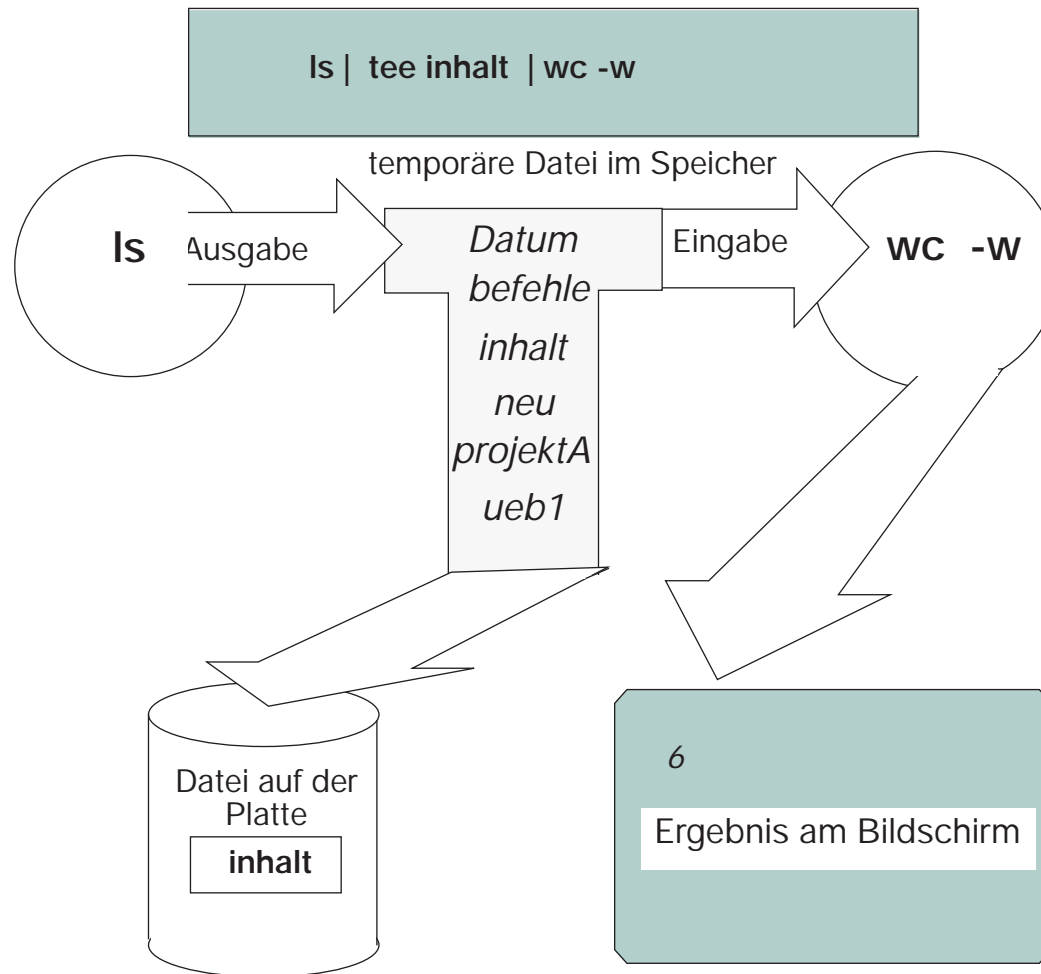
ohne Angabe werden die letzten 10 Zeilen gezeigt

*following*

Wird die Datei gerade erstellt, können Sie hiermit verfolgen, welche Zeilen hinzukommen

um die letzten Zeilen einer Datei anzusehen

# Datenstrom mit tee duplizieren



# Dateien vergleichen

```
diff datei1 datei2
```

*diff* differential file compare

Datei1 h1:

```
hallo  
dies ist die  
erste Datei
```

Datei2 h2:

```
dies ist die  
zweite Datei
```

```
$ diff h1 h2
```

```
1d0
```

```
<hallo
```

```
3c2
```

```
<erste Datei
```

```
---
```

```
>zweite Datei
```

Zeile 1 **d**eletere zu Zeile 0

Inhaltsanzeige

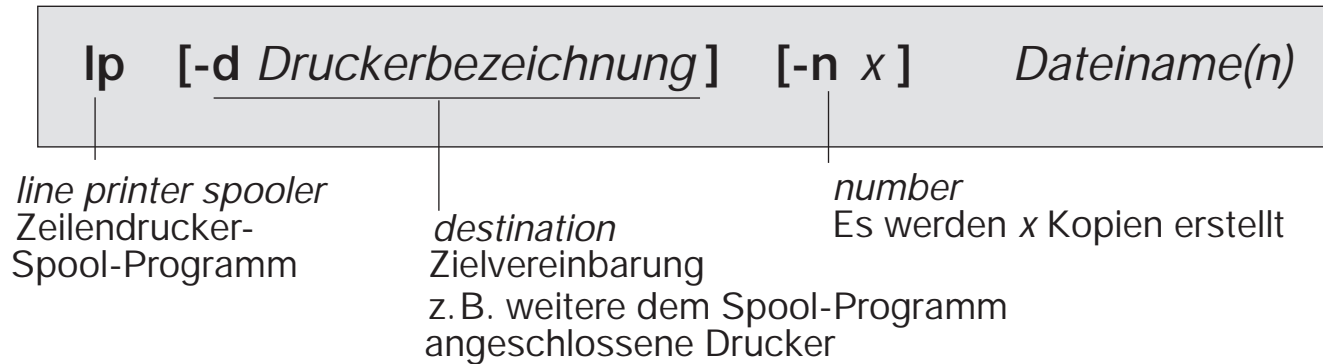
Zeile 3 **c**hange zu Zeile 2

Anzeige h1

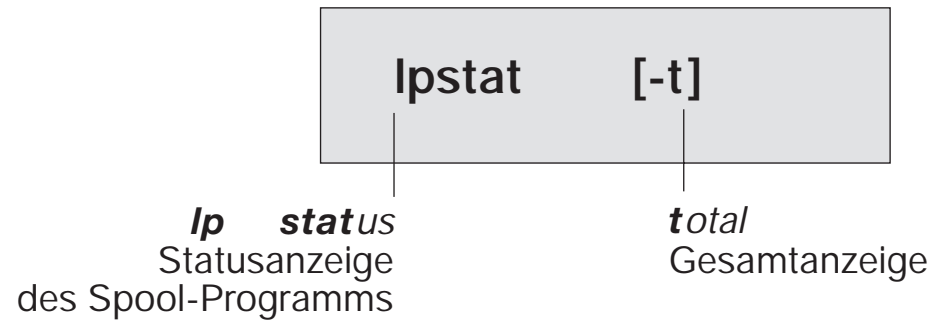
```
---
```

Anzeige h2

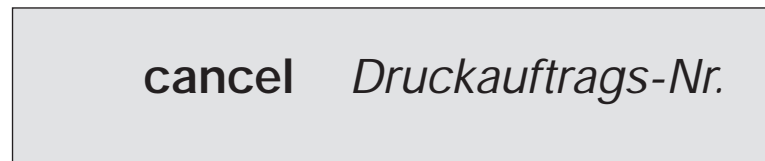
# Dateien drucken



## Kontrolle über Druckaufträge



## Abbruch von Druckaufträgen



# Alternative Druckkommandos

**lpr** *Dateiname(n)*

*line printer*

**lpq**

*line printer queue*

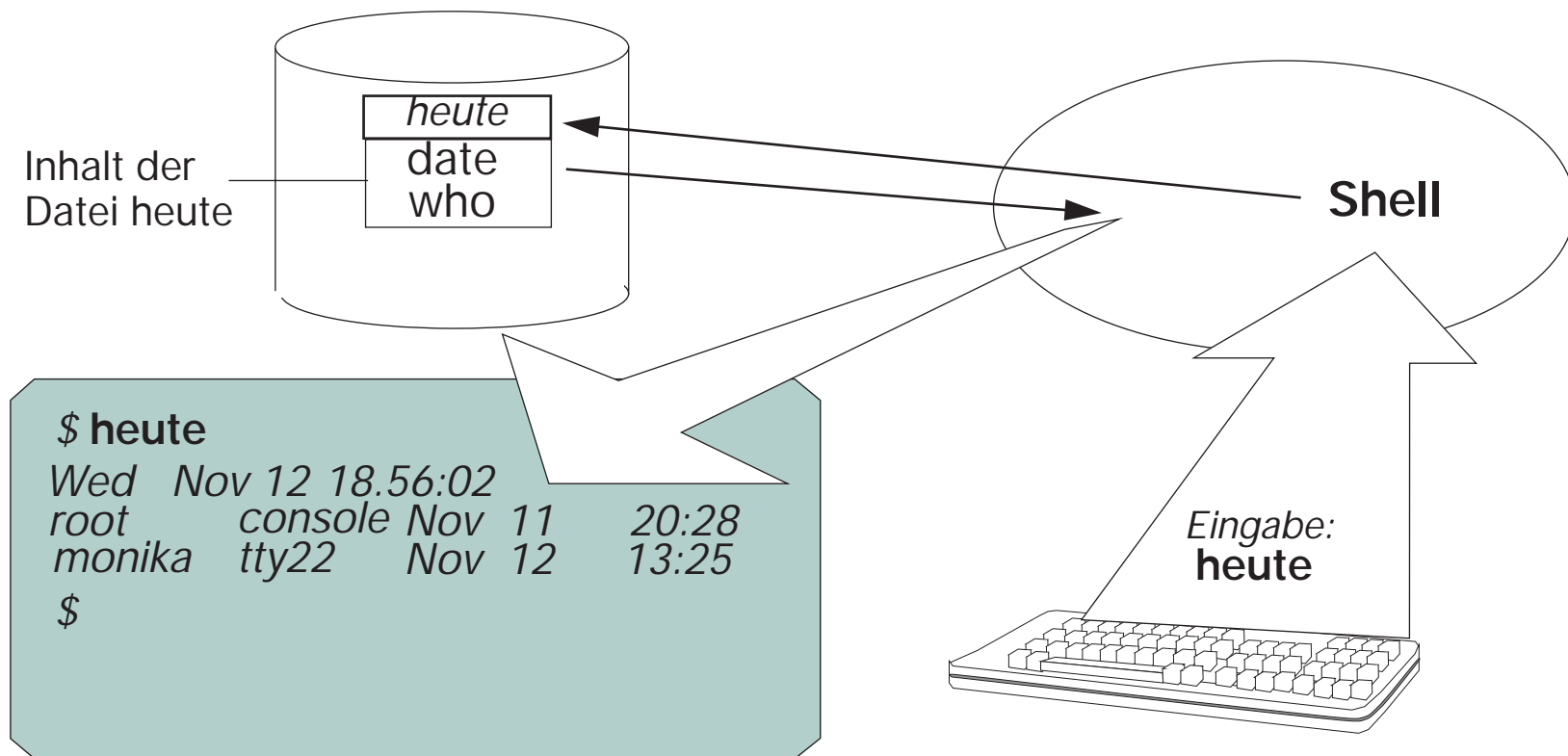
**lprm** *Auftragsnummer*

*line printer remove*

# 12 Shell-Prozeduren

## Was ist ein Shell-Skript (Shell-Prozedur)?

- Nicht anderes als eine ausführbare Datei mit gültigen Kommandos



# Beispiel1

*append* – anhängen — **a** nicht sichtbar

Eingabemodus: **date**

Beenden der Eingabe **who** **ESC**

Beenden vom vi — **ZZ** nicht sichtbar

```
$ mkdir Befehle
$ cd Befehle
$ vi heute
1. Erstellung der Datei mit einem Editor
a nicht sichtbar
date
who ESC
ZZ nicht sichtbar
2. Ändern der Zugriffsrechte
$ chmod +x heute
3. Aufruf der Shell-Prozedur
$ heute
Anzeige von date
Wed Nov 12 19:30:22
root console Nov 11 20:28
monika tty22 Nov 12 13:25
Anzeige von von who
```

# Wie erstellen Sie eigene Kommandos?

- ❑ Über einen Editor  
Kommandozeilen statt über das Terminal in eine Datei eingeben  
mit allen Möglichkeiten z. B.:

- Dateinamenexpansion (Suchmuster/Metazeichen ? \* [ ] [! ^])
- Umleitungen (>, >>, 2>, <, << wort)

Zusätzlich u. a.

- Kommentarzeichen #
- evtl. Run-Kommando bei unterschiedlichen Shells: #!/bin/ksh
- Positionsparameter \$1, \$2, ...
- Variable \$...
- Nachrichten mit echo
- Lesen vom Bildschirm mit read

- ❑ **chmod +x**

# Eigene Kommandos

sind gleichberechtigt mit UNIX-Kommandos

und können beliebig mit anderen Kommandos verknüpft werden  
(Pipelining, Umleitungen usw.)

## Beispiel 2

```
$ vi ls3
```

```
a
```

```
# Ausgabe einer 3-spaltigen Liste
```

```
# seitenweise
```

```
ls | pr -3 -l20 | more ESC
```

```
ZZ
```

```
$ chmod +x ls3
```

```
$ ls3
```

```
PCS          klein          Src  
Befehle      ls3              Uebung  
gross
```

```
$
```

Zeichen für Kommentarzeile in Shell-Prozeduren

# Ergänzen des Suchpfades

- ❑ Um Ihre eigenen Kommandos von jeglicher Position aufrufen zu können, ergänzen Sie bitte in `~/.profile` die Zeile z.B.:

Ergänzung mit:

`PATH=$PATH:$HOME/bin:/home/kurs/bin :$HOME/Befehle`

- ❑ Richten Sie in Ihrem HOME-Directory das Directory *Befehle* ein
- ❑ Erstellen Sie das Kommando **ls3** in Directory **Befehle**
- ❑ Melden Sie sich ab
- ❑ Melden Sie sich wieder an
- ❑ Gehen Sie nach `/usr/bin` und rufen Sie Ihr Kommando **ls3** auf

## Wie verwenden Sie Positionsvariable?

- Ergänzen Sie den Inhalt der Prozedurdatei "ls3" mit:

```
ls      | pr -3 -l20 | more
  ↑
ls $1 | pr -3 -l20 | less
```

und rufen Sie auf:

```
$ ls3 /etc
  ↑
  $1
```

## Welche Positionsparameter gibt es?

**\$0** Name der aufgerufenen Prozedur

**\$1** 1. Parameter des Aufrufes

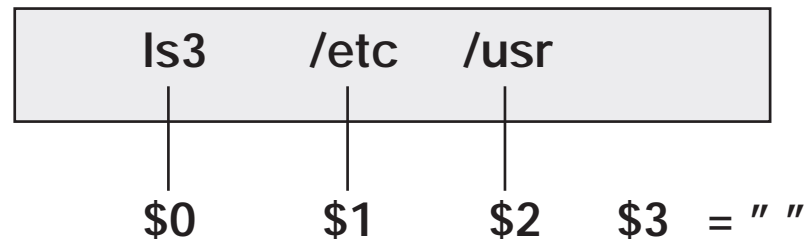
**\$2** 2. Parameter des Aufrufes

...

**\$9** 9. Parameter des Aufrufes

**\$\*** alle Parameter des Aufrufes

**\$#** Anzahl der Parameter



**\$#** = 2

**\$\*** = "/etc /usr"

# Dialog mit dem Benutzer

## Gestalten Sie Ihre Kommandos bedienerfreundlich!

- ❑ Informationen über die Verarbeitung ausgeben (mit echo oder print)
- ❑ Den Benutzer einbeziehen, Fragen stellen
- ❑ Fehler ausschalten
- ❑ Mit read Antworten lesen

**read** *Name einer Variablen (bzw. mehrere)*

*lesen*

## Beispiel 3

# Ausgabe einer 3-spaltigen Liste  
# und Anzeige von Dateiinhalten

```
ls      $1 | pr -3 -l20 | more
```

echo "Welche Datei wollen Sie sehen?"

```
read Datei  
more $Datei
```

← Alle Eingaben des Benutzers bis zum Zeilenende werden der Variablen Datei zugewiesen

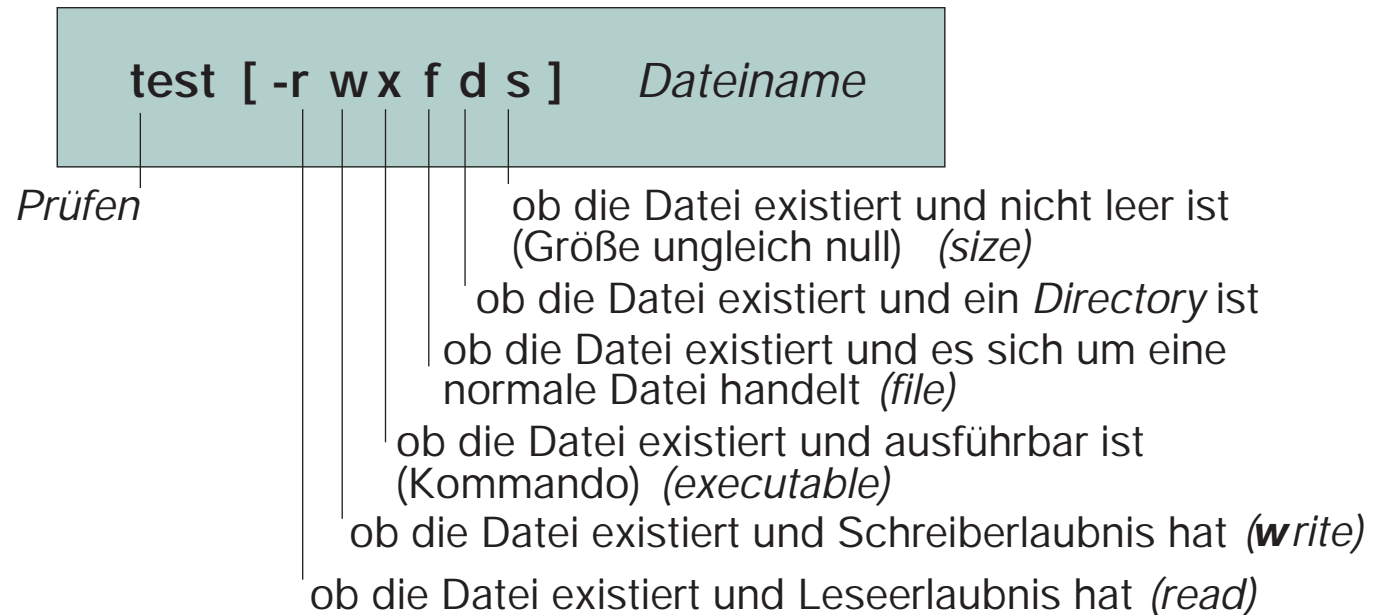
↑ Wird ersetzt durch die Antwort des Benutzers

# 13 Überprüfungen von Bedingungen

## Das test-Kommando

- ❑ Dateitest: prüfen
  - ob eine **Datei existiert**,  
bzw. auf einen bestimmten **Dateityp** abfragen  
bzw. welche **Zugriffsrechte** gesetzt sind
- ❑ Zeichenkettentest: prüfen
  - ob **Zeichenfolgen** (*Strings*) vorhanden sind
- ❑ Wertetest: vergleichen von **Zeichenfolgen**
  - auf Textgleichheit
  - wenn es sich um Zahlen handelt:  
auf algebraische Vergleiche  
(gleich, größer gleich, kleiner, kleiner gleich, nicht gleich)

# Dateitest



Es kann jeweils nur eine Option abgefragt werden  
oder eine Verknüpfung mit -o (oder) bzw. -a (and) erfolgen

Beispiel: `test -f $Datei -a -w $Datei`

## Beispiel 4 - Prüfen auf Vorhandensein

```
# Ausgabe einer 3-spaltigen Liste  
# und Anzeige von Dateiinhalten
```

```
ls      $1 | pr -3 -l20 | more
```

```
echo "Welche Datei wollen Sie sehen?"  
read Datei  
test -f $Datei && more $Datei
```

wenn cmd1 erfolgreich ist, führe cmd2 aus

# Zeichenketten- und Wertetest

- ❑ Ist eine Zeichenkette vorhanden?

```
test [ -n ] Zeichenkette
```

Prüfen

**n** (not zero) ob die Zeichenkette nicht leer ist  
oder **z** (zero) ob die Zeichenkette leer ist

- ❑ Vergleich von Texten

```
test Zeichenkette-A = Zeichenkette-B
```

Prüfen

**Beachten Sie, daß unbedingt Leerzeichen  
zwischen den einzelnen Feldern stehen müssen!**

## Beispiel 5

```
# Ausgabe einer 3-spaltigen Liste  
# und Anzeige von Dateiinhalten
```

```
ls      $1| pr -3 -l20 | more  
echo "Welche Datei wollen Sie sehen?"  
read Datei  
test -f $Datei && more $Datei
```

```
echo "Soll die Datei gelöscht werden? (j/n)"  
read Antw  
test "$Antw" = "j" && rm -i $Datei
```

## test - moderne Schreibweise

```
[ Zeichenkette-A = Zeichenkette-B ]
```

Beachten Sie, daß unbedingt Leerzeichen zwischen den einzelnen Feldern sein müssen!

## Kombination von test

```
test 1. Vergleich [ -a ] 2. Vergleich [ weitere Verknüpfungen ]
```

(and) und-Verknüpfung, beide Abfragen müssen erfüllt sein  
(or) oder-Verknüpfung, eine von beiden Abfragen muß erfüllt sein.

# Algebraische Vergleiche

| Abfrage auf                                  | Kommandoeingabe                            | alternative Schreibweise           |
|--|--|------------------------------------|
| <b>equal</b><br><i>gleich</i>                | <b>test</b> <i>n1</i> <b>-eq</b> <i>n2</i> | [ <i>n1</i> <b>-eq</b> <i>n2</i> ] |
| <b>not equal</b><br><i>nicht gleich</i>      | <b>test</b> <i>n1</i> <b>-ne</b> <i>n2</i> | [ <i>n1</i> <b>-ne</b> <i>n2</i> ] |
| <b>less than</b><br><i>kleiner als</i>       | <b>test</b> <i>n1</i> <b>-lt</b> <i>n2</i> | [ <i>n1</i> <b>-lt</b> <i>n2</i> ] |
| <b>less equal</b><br><i>kleiner gleich</i>   | <b>test</b> <i>n1</i> <b>-le</b> <i>n2</i> | [ <i>n1</i> <b>-le</b> <i>n2</i> ] |
| <b>greater than</b><br><i>größer als</i>     | <b>test</b> <i>n1</i> <b>-gt</b> <i>n2</i> | [ <i>n1</i> <b>-gt</b> <i>n2</i> ] |
| <b>greater equal</b><br><i>größer gleich</i> | <b>test</b> <i>n1</i> <b>-ge</b> <i>n2</i> | [ <i>n1</i> <b>-ge</b> <i>n2</i> ] |

## Beispiel 6

Teil einer Shell-Prozedur mit Abfrage der Anzahl der Parameter

```
if test "$#" -lt "2"
```

Wenn die Anzahl der Parameter kleiner als 2 ist, dann ...

```
    then echo "Es müssen mindestens 2 Parameter angegeben werden"  
        exit
```

```
fi
```

# 14 Ablaufsteuerung

## Programmiersprachenkonstrukte

- ❑ Prüfungen mit dem Kommando test
- ❑ Bedingungen z. B. wenn das vorherige Kommando erfolgreich war, dann bzw. nicht erfolgreich war

```
cmd1 && cmd2  
cmd1 || cmd2
```

- ❑ if-Bedingung

```
if cmd ...  
    then cmd ...  
    else cmd ...  
fi
```

- ❑ case-Bedingungen
- ❑ Schleifen mit

```
case ... - esac
```

```
for ...  
    do ...  
done
```

oder mit

```
while ...  
    do ...  
done
```

## Die if-Abfrage

|   |                           |
|---|---------------------------|
| <b>if</b> <i>Kommando (Bedingung)</i> _____ | wenn das und das zutrifft |
| <b>then</b> <i>Kommandofolge1</i> _____     | dann tue ....             |
| [ <b>else</b> <i>Kommandofolge2</i> ] _____ | [ sonst tue ... ]         |
| <b>fi</b> _____                             | fertig (Ende der Abfrage) |

|   |   |
|---|---|
| ...   |   |
| <b>read</b> <i>Datei</i>                                    |   |
| <b>if</b> <b>test -f</b> <i>\$Datei</i> _____               | Wenn die angegebene Datei ( <i>\$Datei</i> ) existiert und es sich um eine normale Datei handelt ( <i>test -f</i> ) |
| <b>then</b> <b>pr -n</b> <i>\$Datei</i>   <b>more</b> _____ | dann soll sie am Bildschirm angezeigt werden ( <i>more</i> )  |
| <b>else</b> <b>echo "\$Datei ist keine Datei"</b> _____     | sonst soll eine entsprechende Nachricht ausgegeben und die Prozedur beendet werden ( <i>echo; exit</i> )            |
| <b>exit</b>   |   |
| <b>fi</b> _____   | Ende (fi) Ende der Abfrage  |

## Die case-Anweisung

**case** *Wort* **in**

*Muster1* ) *Kommandoliste1*;;

*Muster2* | *Muster3* ) *Kommandoliste2*;;

...

2 Semikolons als Abschluß der Kommandoliste

**esac**

Falls die Zeichenkette ›Wort‹ mit Muster1 übereinstimmt, dann tue (Kommandoliste1)  
stimmt sie mit Muster2 oder Muster3 überein, führe die Kommandoliste aus

Ende der ›case-Bedingung‹

...

**read** *Antw*

**case** *\$antw* **in**

*j* | *J* ) *rm \$Datei* ;;

*n* | *N* ) *echo \$Datei wurde nicht gelöscht* ;;

*\** ) *echo "nur j oder n als Eingabe erlaubt"* ;;

**esac**

Falls *in* Antwort (*\$antw*)  
das Muster *j* oder *J* enthalten ist,  
lösche die Datei (*\$Datei*)

das Muster *n* oder *N* enthalten ist,  
gebe eine Nachricht aus (*echo*)

alle anderen Muster  
gebe Hinweis aus (*echo*)

Ende Ende der Abfrage

# Die for-Schleife

|             |                           |           |                    |   |                          |
|-------------|---------------------------|-----------|--------------------|---|--------------------------|
| <b>for</b>  | <i>Name</i>               | <b>in</b> | <i>Wort1 Wortn</i> | — | Wert, der der Variablen  |
|             |                           |           |                    | — | Name zugeordnet wird     |
| <b>do</b>   |                           |           |                    |   | Solange ein Wert da ist, |
|             | <i>Kommandoliste ....</i> |           |                    |   | tue                      |
| <b>done</b> |                           |           |                    |   | Ende der Schleife        |

```
for   Name _____  
  
      do       Kommandoliste ....  
  
      done
```

Name der Variablen, die  
nacheinander den Wert  
von \$1 -\$n übernimmt  
Solange ein Wert da ist,  
tue  
fertig – gehe zu Beginn  
der Schleife

# Die while- und until-Schleife

|              |                           |  |
|--------------|---------------------------|--|
| <b>while</b> | <i>Kommando</i>           | Solange das Kommando »wahr«<br>ist ( <i>Exit-Status 0 liefert</i> ),<br>tue ...<br><br>fertig – gehe zu Beginn<br>der Schleife |
| <b>do</b>    | <i>Kommandoliste ....</i> |  |
| <b>done</b>  |                           |  |

|              |                 |   |
|--------------|-----------------|---|
| <b>until</b> | <i>Kommando</i> | Solange das Kommando ›unwahr‹<br>(Exit-Status ungleich 0) liefert –<br>nicht erfolgreich war,<br><i>do</i> <i>Kommandoliste ....</i><br><i>do</i> <i>tue ...</i><br><b>done</b> fertig – gehe zu Beginn<br>der Schleife |
|              |                 |   |
|              |                 |   |

## Beispiel einer kleinen Shell-Prozedur

```
#!/bin/ksh
# Prozedur um einen Countdown auf dem Bildschirm auszugeben
typeset -i Zahl ←————— Definition der Variablen als Integer
echo " Mit welcher Zahl soll der Countdown starten?"
Bitte geben Sie eine Zahl ein"
read Zahl
clear
echo "Der Countdown startet"
while test $Zahl -gt 0
do clear
echo $Zahl
Zahl=$Zahl-1 ←————— Direkte Rechenoperation mit der Integer-Variablen
done
```

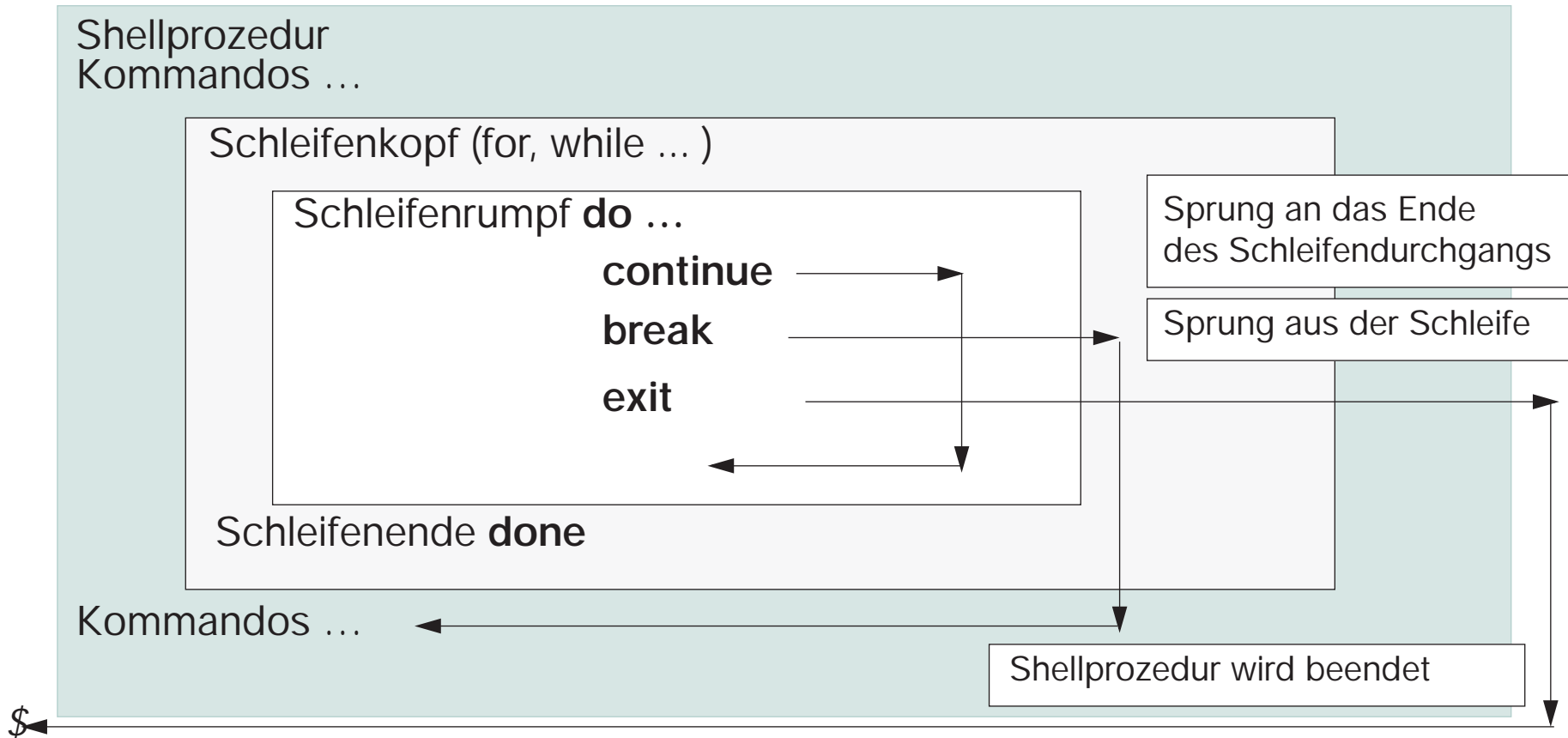
**clear** - Kommando, um den aktuellen Bildschirminhalt zu löschen

**banner** - Kommando, um Zeichenketten im Großformat auszugeben  
nicht auf allen Unix-Systemen verfügbar

Direkte Rechenoperation mit der Integer-Variablen

# Vorzeitiges Beenden von Shellprozeduren

*\$ Aufruf einer Shellprozedur*



## Einige Shell-interne Kommandos

| Eingabe                    | Auswirkung  |
|----------------------------|---|
| <b># <i>Text</i></b>       | Kommentar bis Ende der Zeile  |
| <b>#!/bin/ksh</b>          | run-Kommando, die Prozedur wird von der Korn-Shell abgearbeitet   |
| <b>. <i>datei</i></b>      | Liest an dieser Stelle die Datei <i>datei</i> ein   |
| <b>break</b>               | Sprung aus der Schleife (for, while, until)   |
| <b>cd <i>Directory</i></b> | Wechselt in das angegebene Directory (gilt nur während der Prozedur - nach Beendigung wieder im gleichen Directory wie beim Aufruf) |
| <b>continue</b>            | Sprung an das Ende einer Schleife   |
| <b>exit [<i>n</i>]</b>     | Prozedur wird beendet (ungleich 0, bzw. entsprechende angegebene Zahl)  |

| Eingabe                             | Auswirkung  |             |         |             |          |     |     |            |       |       |              |     |     |
|-------------------------------------|---|-------------|---------|-------------|----------|-----|-----|------------|-------|-------|--------------|-----|-----|
| <b>export</b> <i>Variable</i>       | Variable mit der Wertzuweisung wird an evtl. Unterprozesse weitergeleitet   |             |         |             |          |     |     |            |       |       |              |     |     |
| <b>read</b> <i>name [name2 ...]</i> | liest von der Standardeingabe und weist die Worte den angegebenen Variablen <i>name name2</i> zu (letzte der angegebenen Variable wird der Rest bis Zeilenende zugewiesen)  |             |         |             |          |     |     |            |       |       |              |     |     |
| <b>set -x</b>                       | zeigt die Ausführung jeder Kommandos ( <i>execution</i> )   |             |         |             |          |     |     |            |       |       |              |     |     |
| <b>set -</b>                        | setzt die -x Option wieder aus  |             |         |             |          |     |     |            |       |       |              |     |     |
| <b>umask</b> <i>nnn</i>             | Setzt die Maske für die Zugriffswerte (Minuswert zu Oktalzahl) <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">Dateien</th> <th style="text-align: center;">Directories</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Standard</td> <td style="text-align: center;">666</td> <td style="text-align: center;">777</td> </tr> <tr> <td style="text-align: center;">z.B. umask</td> <td style="text-align: center;">002 -</td> <td style="text-align: center;">002 -</td> </tr> <tr> <td style="text-align: center;">neuer Modus:</td> <td style="text-align: center;">664</td> <td style="text-align: center;">775</td> </tr> </tbody> </table> |             | Dateien | Directories | Standard | 666 | 777 | z.B. umask | 002 - | 002 - | neuer Modus: | 664 | 775 |
|                                     | Dateien   | Directories |         |             |          |     |     |            |       |       |              |     |     |
| Standard                            | 666   | 777         |         |             |          |     |     |            |       |       |              |     |     |
| z.B. umask                          | 002 -   | 002 -       |         |             |          |     |     |            |       |       |              |     |     |
| neuer Modus:                        | 664   | 775         |         |             |          |     |     |            |       |       |              |     |     |

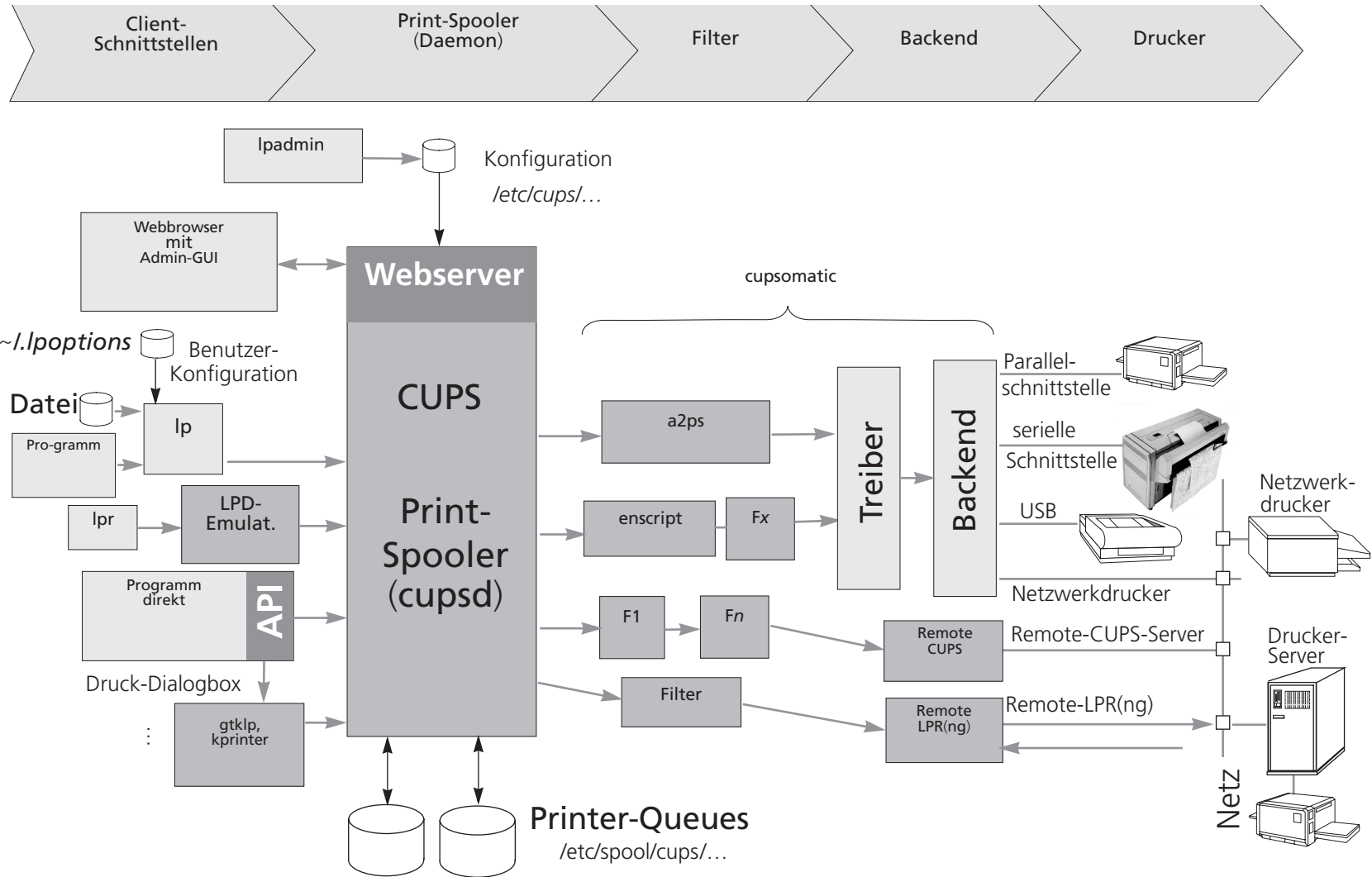
# 15 Drucken unter Linux

- ❑ Print-Spooler-System
- ❑ Postscript - Die Druckersprache unter UNIX
- ❑ Filter (Umsetzen in Postscript - z. B. Ghostscript)  
UNIX erkennt in der Regel das jeweilige Dateiformat  
und wandelt in Postscript um
- ❑ Mehrere Druckerwarteschlangen (Queue)
- ❑ Das CUPS (*Common Unix Printing System*)

# Verwalten von Druckaufträgen

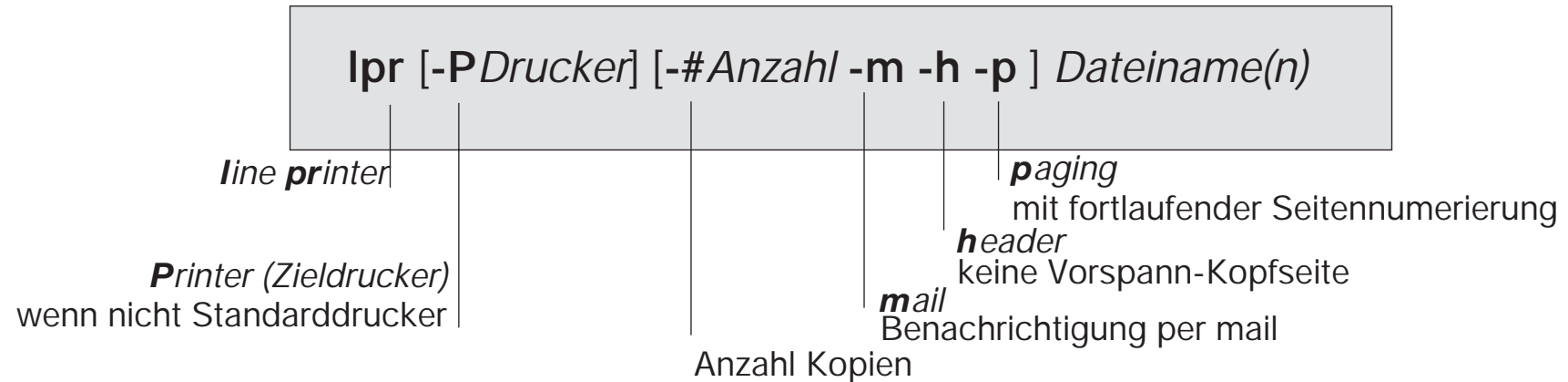
- Zwei Spoolingsysteme in CUPS vereint:
  - **lpsched** (SystemV: Solaris, HP-UX, AIX u.a.)  
mit **lp** (Druckauftrag), **cancel** (Auftrag löschen) und **lpstat** (Druckstatus)
  - **lpd** (Berkley: (Linux, Digital UNIX, AIX)  
mit **lpr** (Druckauftrag), **lprm** (Auftrag löschen), **lpq** (Druckanzeige *queue*)

# Topologie des CUPS-Systems

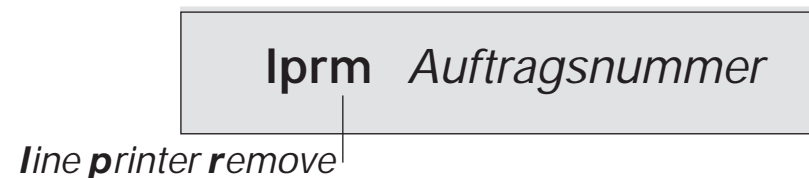


# Druckkommandos lpr

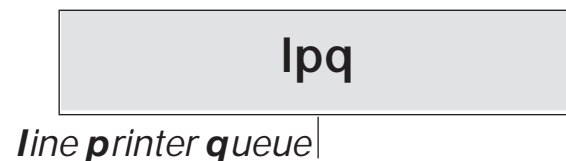
## □ Druckauftrag starten



## □ Druckauftrag löschen



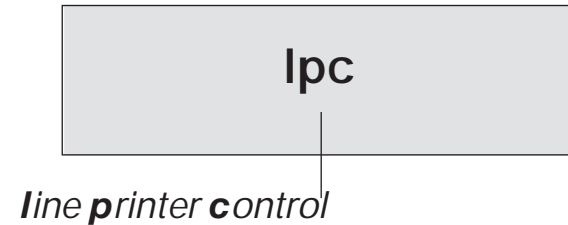
## □ Druckaufträge anzeigen



# Kontrolle mit lpc

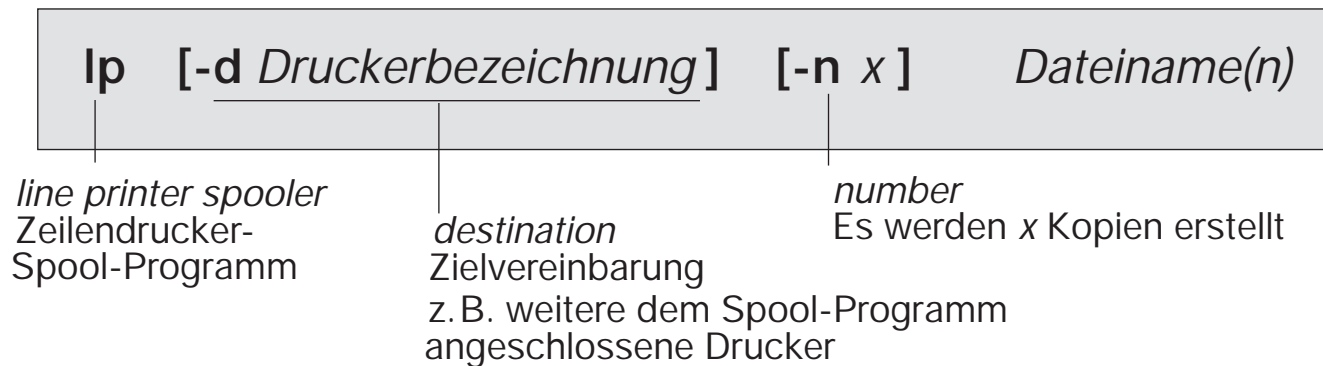
## □ Druckaufträge kontrollieren

- Drucker aktivieren  
**enable drucker**
- Drucker deaktivieren  
**disable drucker**
- Alle Druckaufträge eines Druckers löschen  
**clear drucker**
- Reihenfolge verändern  
**topq drucker Auftragsnummer (job-ID)**  
**topq drucker username**
- Statusinformationen über Drucker  
**status drucker**



# Druckkommandos Ip

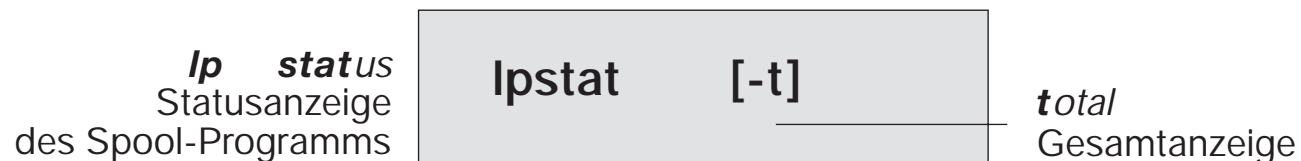
## ❑ Druckauftrag starten



## ❑ Druckauftrag löschen



## ❑ Druckaufträge Status abfragen



# 16 Systemüberwachung

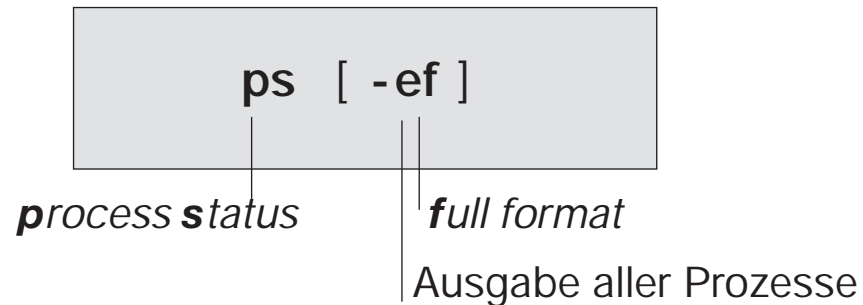
## Jobcontrol - Prozeßverwaltung

Was ist ein Prozeß?

Ein gestartetes Programm

# Wie sehen Sie, welche Prozesse vorhanden sind?

ps – Kommando, um den Status der Programme abzufragen



`ps [-efl] [-u Benutzer]`  
`ps -[axl]`

*process status*

**Anzeige der aktuellen Prozesse**

**-e every** Anzeige aller Prozesse

**-f full** volles Format

**-l long** mit allen Attributen

**-u user** Prozesse des angegebenen Benutzers

Bei einigen Systemen werden unterschiedliche Optionen verlangt so z. B.

**-a all** alle Prozesse eines Terminals

**-x** alle Systemprozesse

```
$ ps -ef
```

| <i>UID</i> | <i>PID</i> | <i>PPID</i> | <i>STIME</i> | <i>TTY</i> | <i>TIME</i> | <i>COMMAND</i>        |
|------------|------------|-------------|--------------|------------|-------------|-----------------------|
| root       | 0          | 0           | Jun          | ?          | 0:05        | swapper               |
| root       | 1          | 0           | Jun          | ?          | 0:02        | /etc/init             |
| monika     | 23         | 1           | 8:30:01      | tty11      | 0:03        | -sh                   |
| gisela     | 24         | 1           | 8:45:31      | tty13      | 0:03        | -sh                   |
| hans       | 25         | 1           | 8:55:02      | tty12      | 0:02        | -sh                   |
| monika     | 26         | 23          | 9:02:00      | tty11      | 0:04        | sort /usr/lib/cookies |
| monika     | 28         | 23          | 9:02:30      | tty11      | 0:01        | ps -ef                |
| gisela     | 29         | 24          | 9:02:10      | tty13      | 0:01        | ls /bin               |
| gisela     | 30         | 24          | 9:02:10      | tty13      | 0:01        | wc -l                 |

Terminal

Kommandoaufruf

Startzeit

Zeit in Minuten und Sekunden

*Parent Process Identification*  
Vaterprozeß

*Process Identification*  
Prozeßnummer

*User Identification*  
Benutzername

## Wie können Sie bestimmte Prozesse auswählen?

- ❑ Mit Hilfe einer Pipe und dem Kommando grep "Muster"
- ❑ Als Muster können beliebige Namen wie Prozeßname, UID etc. abgefragt werden

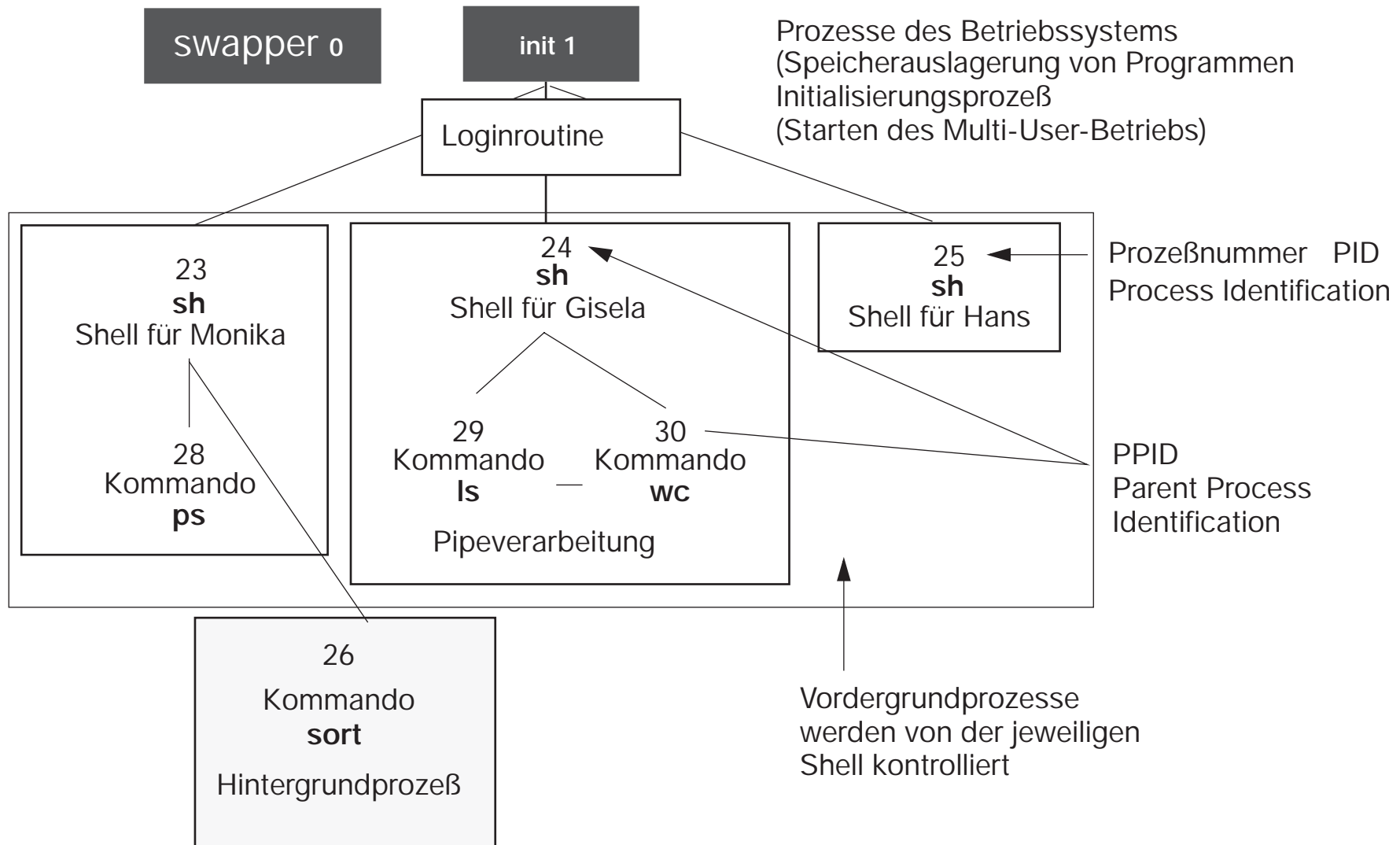
```
$ ps -ef | grep sh
```

| <i>UID</i>    | <i>PID</i> | <i>PPID</i> | <i>STIME</i>   | <i>TTY</i>   | <i>TIME</i> | <i>COMMAND</i> |
|---------------|------------|-------------|----------------|--------------|-------------|----------------|
| <i>monika</i> | <i>23</i>  | <i>1</i>    | <i>8:30:01</i> | <i>tty11</i> | <i>0:03</i> | <i>-sh</i>     |
| <i>gisela</i> | <i>24</i>  | <i>1</i>    | <i>8:45:31</i> | <i>tty13</i> | <i>0:03</i> | <i>-sh</i>     |
| <i>hans</i>   | <i>25</i>  | <i>1</i>    | <i>8:55:02</i> | <i>tty12</i> | <i>0:02</i> | <i>-sh</i>     |

# Prozeßattribute

|                  |  |
|------------------|--|
|                  |  |
| <b>USER, UID</b> | Der Benutzer, welchem der Prozeß gehört bzw. dessen UID. System-prozesse gehören immer dem Benutzer root und dienen der Verwaltung des Systems und des Netzwerkes.   |
| <b>SZ</b>        | Die Größe des virtuellen Arbeitsspeichers in KB  |
| <b>PPID</b>      | Die PID des Elternprozesses  |
| <b>STIME</b>     | Die Zeit, zu der der Prozeß gestartet wurde  |
| <b>PRI</b>       | Die Priorität des Prozesses (je niedriger die Zahl, desto höher die Priorität)   |
| <b>TIME</b>      | Die bisher verbrauchte CPU-Zeit  |
| <b>STAT, S</b>   | <p>Der Zustand des Prozesses:</p> <p><b>R</b> (running): Der Prozeß wird gerade von der CPU bearbeitet</p> <p><b>I</b> (idle): Der Prozeß ist längere Zeit untätig</p> <p><b>Z</b> (zombie): Der Prozeß hat terminiert, kann aber nicht aus der Prozeßta-belle gelöscht werden.</p> <p><b>S</b> (sleeping): Der Prozeß ist kurzzeitig untätig oder wartet auf ein Ereignis</p> |

# Vereinfachte Darstellung der Prozeß-Hierarchie

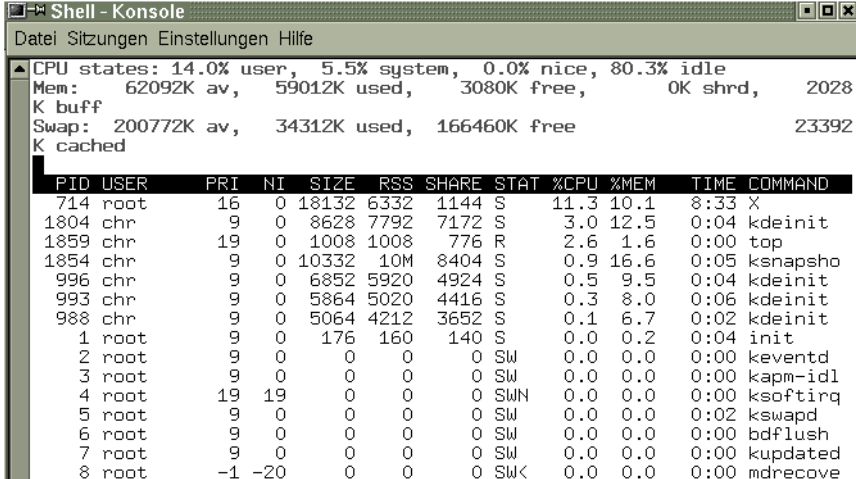


# Weitere Kommandos zur Anzeige von Prozessen

## Über pstree

```
knotify
chr@J0GYli:~> pstree
init--afpd
|--artsd
|--atalkd
|--atd
|--cron
|--inetd
|--kapm-idled
|--5*[kdeinit]
|--kdeinit--3*[kdeinit]
|   |--kdeinit---bash
|   |--kdeinit--bash---man---sh---less
|   |--kdeinit--bash---pstree
|   |--kdesu---su---kdesu_stub---yast2---y2bignfat---y2bignf:
|--kdeinit---cat
|--kdesud,
```

## Anzeiger der Top-aktuellen Prozesse über top



```
Shell - Konsole
Datei Sitzungen Einstellungen Hilfe
^ CPU states: 14.0% user, 5.5% system, 0.0% nice, 80.3% idle
Mem: 62092K av, 59012K used, 3080K free, 0K shrd, 2028
K buff
Swap: 200772K av, 34312K used, 166460K free 23392
K cached
PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM TIME COMMAND
714 root 16 0 18132 6332 1144 S 11.3 10.1 8:33 X
1804 chr 9 0 8628 7792 7172 S 3.0 12.5 0:04 kdeinit
1859 chr 19 0 1008 1008 776 R 2.6 1.6 0:00 top
1854 chr 9 0 10332 10M 8404 S 0.9 16.6 0:05 ksnapsho
996 chr 9 0 6852 5920 4924 S 0.5 9.5 0:04 kdeinit
993 chr 9 0 5864 5020 4416 S 0.3 8.0 0:06 kdeinit
988 chr 9 0 5064 4212 3652 S 0.1 6.7 0:02 kdeinit
1 root 9 0 176 160 140 S 0.0 0.2 0:04 init
2 root 9 0 0 0 0 SW 0.0 0.0 0:00 keventd
3 root 9 0 0 0 0 SW 0.0 0.0 0:00 kapm-idl
4 root 19 19 0 0 0 SWN 0.0 0.0 0:00 ksoftirq
5 root 9 0 0 0 0 SW 0.0 0.0 0:02 kswapd
6 root 9 0 0 0 0 SW 0.0 0.0 0:00 bdflush
7 root 9 0 0 0 0 SW 0.0 0.0 0:00 kupdated
8 root -1 -20 0 0 0 SWK 0.0 0.0 0:00 mdrecove
```

# Prioritäten verändern

Beim Start eines Kommandos mit

```
nice [-nice-Wert] [Kommando [Argumente ...]]
```

von -20 bis +19

(höchste bis niedrigste)

negative Werte nur durch Systemverwalter

nice-Wert **-10** ist positiv

nice-Wert **--10** ist negativ

❑ Nachträglich mit

```
renice oder snice neuer-nice-Wert [[-p] PID ...] [[-u] user ...]
```

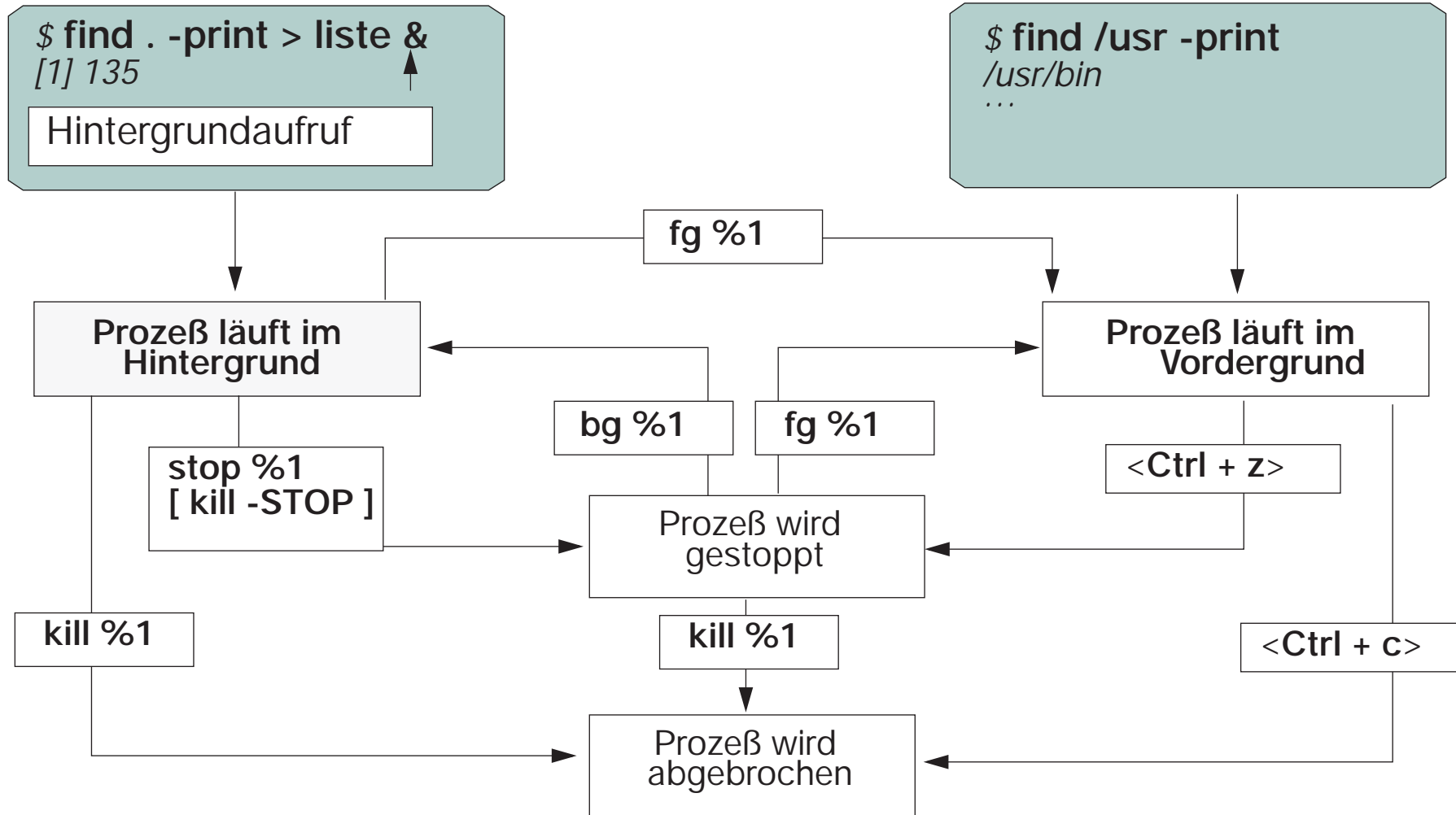
```
# renice -5 123 -u daemon bin root -p 32
```

In diesem Beispiel wird der nice-Wert der Prozesse mit den PIDs 32 und 123, sowie alle Prozesse, die den Benutzern daemon, bin und root gehören, um 5 reduziert

renice-Wert **10** ist positiv

renice-Wert **-10** ist negativ

# Wie können Sie Prozesse steuern?



## Beispiel einer Jobsteuerung

```
/usr/kurs/hans > find . -print > liste &  
[1] 135  
/usr/kurs/hans > stop %1  
[1] stopped  
/usr/kurs/hans > fg %1
```

# Kommandos zur Prozeßsteuerung

`ps -ef | grep Benutzernamen (oder Programmnamen)`

|  |  |
|--|--|
| <b>jobs</b>  | <i>Jobcontrol nur ksh/csh</i><br><b>Zeigt vorhandene Jobs an</b><br>Jobs sind gestoppte Vordergrundprozesse ( <b>Ctrl+z</b> ) oder Hintergrundprozesse<br>› <b>fg</b> › <b>bg</b>  |
| <b>kill -9 [PID] [%Jobnr]</b><br>einige Signale:<br>1 HUP<br>2 INT<br>3 QUIT<br>9 KILL<br>17 STOP<br>18 TSTP | <i>kill (töten)</i><br><b>Bricht einen Prozeß sicher ab</b><br>Beendet Terminalverbindung<br>Entspricht Ctrl+c Canceln<br>Abbruch mit coredump<br>Absoluter Prozeßabbruch<br>Stoppt einen Prozeß (stop)<br>Stoppt einen Vordergrundprozeß (Ctrl+z) |

# Kommandos zur Prozeßsteuerung

## Fortsetzung

|   |   |
|---|---|
| <b>bg</b> %Jobnummer                        | <i>background</i> <span style="float: right;"><i>nur ksh/bash</i></span><br><b>Der Job läuft als Hintergrundprozeß weiter</b><br><br>› <b>fg</b> › <b>stop</b>  |
| <b>fg</b> %Jobnummer                        | <i>foreground</i> <span style="float: right;"><i>nur ksh/bash/csh</i></span><br><b>Der Job läuft als Vordergrundprozeß weiter</b><br>Gilt nur für Hintergrund- oder gestoppte Prozesse<br>› <b>bg</b> › <b>stop</b> |
| <b>stop</b> %Jobnummer<br><b>kill -STOP</b> | <i>anhalten</i> <span style="float: right;"><i>nur ksh/bash/csh</i></span><br><b>Stoppen eines Jobs/Prozesses</b><br>(bei Vordergrundprozessen: <b>Ctrl+z</b> )   |

## Wie können Sie mehrere Prozesse killen?

- ❑ Auswahl der Prozesse über Pipe, Weitergabe an eine Schleife mit read

```
# ps -ef | grep sh
```

| <i>UID</i>    | <i>PID</i> | <i>PPID</i> | <i>STIME</i>   | <i>TTY</i>   | <i>TIME</i> | <i>COMMAND</i> |
|---------------|------------|-------------|----------------|--------------|-------------|----------------|
| <i>monika</i> | <i>23</i>  | <i>1</i>    | <i>8:30:01</i> | <i>tty11</i> | <i>0:03</i> | <i>-sh</i>     |
| <i>gisela</i> | <i>24</i>  | <i>1</i>    | <i>8:45:31</i> | <i>tty13</i> | <i>0:03</i> | <i>-sh</i>     |
| <i>hans</i>   | <i>25</i>  | <i>1</i>    | <i>8:55:02</i> | <i>tty12</i> | <i>0:02</i> | <i>-sh</i>     |

```
# ps -ef | grep sh | while read UID PID Rest
```

```
> do kill -9 $PID
```

```
> done
```

```
PID 23 killed
```

```
PID 24 killed
```

```
PID 25 killed
```

# Prozesskontrolle unter der KDE

## System R Info R Prozessverwaltung

Performance-  
anzeige

Auswahl  
Prozessart

Anklicken  
für gewünschte  
Sortierreihen-  
folge

The screenshot shows the KPM window with the following data:

| PID  | USER | PRI | NICE | SIZE  | RSS   | SHARE | STA | %CPU | %MEM  | TIME  | CMDLINE   |
|------|------|-----|------|-------|-------|-------|-----|------|-------|-------|---|
| 828  | chr  | 0   | 0    | 656   | 420   | 416   | S   | 0,00 | 0,68  | 0,56s | /bin/bash --login /usr/X11R6/bin/kde                            |
| 941  | chr  | 9   | 0    | 2492  | 1716  | 1564  | S   | 0,00 | 2,76  | 0,19s | kdeinit: dcopserver --nosid                                     |
| 944  | chr  | 9   | 0    | 3640  | 2908  | 2724  | S   | 0,00 | 4,68  | 0,29s | kdeinit: klauncher  |
| 946  | chr  | 9   | 0    | 4308  | 3240  | 2996  | S   | 0,00 | 5,22  | 1,35s | kdeinit: kded   |
| 952  | chr  | 9   | 0    | 2924  | 2620  | 2260  | S   | 0,00 | 4,22  | 1,13s | /opt/kde2//bin/artsd -F 10 -S 4096 -s 60 -m artsmessage -l 3 -f |
| 953  | chr  | 9   | 0    | 1752  | 992   | 908   | S   | 0,00 | 1,60  | 0,13s | kdeinit: Running...   |
| 966  | chr  | 9   | 0    | 5184  | 4364  | 3804  | S   | 0,00 | 7,03  | 1,00s | knotify   |
| 970  | chr  | 9   | 0    | 3176  | 2364  | 2100  | S   | 0,19 | 3,81  | 0,96s | ksmserver --restore   |
| 971  | chr  | 9   | 0    | 4576  | 3768  | 3136  | S   | 0,58 | 6,07  | 4,27s | kdeinit: kwin -session 11c0a80009000100653073000000019900       |
| 975  | chr  | 9   | 0    | 2784  | 1796  | 1680  | S   | 0,00 | 2,89  | 0,39s | kdeinit: kwrited  |
| 976  | chr  | 9   | 0    | 5408  | 4636  | 3940  | S   | 0,00 | 7,47  | 8,28s | kdeinit: kdesktop   |
| 977  | chr  | 9   | 0    | 400   | 320   | 316   | S   | 0,00 | 0,52  | 0,04s | /bin/cat  |
| 979  | chr  | 9   | 0    | 9052  | 8328  | 6676  | S   | 0,19 | 13,41 | 0,13  | kdeinit: kicker   |
| 982  | chr  | 9   | 0    | 4424  | 3272  | 2896  | S   | 0,00 | 5,27  | 1,15s | kdeinit: konsole -session 104ff0962c000100902642700             |
| 983  | chr  | 9   | 0    | 9504  | 8396  | 6832  | S   | 0,19 | 13,52 | 5,39s | kdeinit: konqueror -session 11c0a8005a000101069050              |
| 984  | chr  | 9   | 0    | 1408  | 1016  | 1012  | S   | 0,00 | 1,64  | 0,13s | /bin/bash   |
| 1399 | chr  | 9   | 0    | 7284  | 6500  | 6024  | S   | 0,00 | 10,47 | 3,35s | kdeinit: konsole -icon konsole -miniicon konsole                |
| 1400 | chr  | 9   | 0    | 1544  | 1288  | 1148  | S   | 0,00 | 2,07  | 0,26s | /bin/bash   |
| 1663 | chr  | 17  | 0    | 7772  | 7764  | 6368  | R   | 4,83 | 12,50 | 0,11  | kpm -icon kpm -miniicon kpm                                     |
| 1665 | chr  | 9   | 0    | 5224  | 4724  | 4444  | S   | 0,00 | 7,61  | 0,25s | kdeinit: kio_file file /tmp/ksocket-chr/klauncherzUg8b.sl       |
| 1666 | chr  | 9   | 0    | 5220  | 4720  | 4440  | S   | 0,00 | 7,60  | 0,14s | kdeinit: kio_file file /tmp/ksocket-chr/klauncherzUg8b.sl       |
| 1711 | chr  | 9   | 0    | 10120 | 10120 | 8288  | S   | 0,58 | 16,30 | 3,47s | ksnapshot -icon ksnapshot -miniicon ksnapshot                   |

Steuerung  
der ausgewählten  
Prozesse über  
Kontextmenü

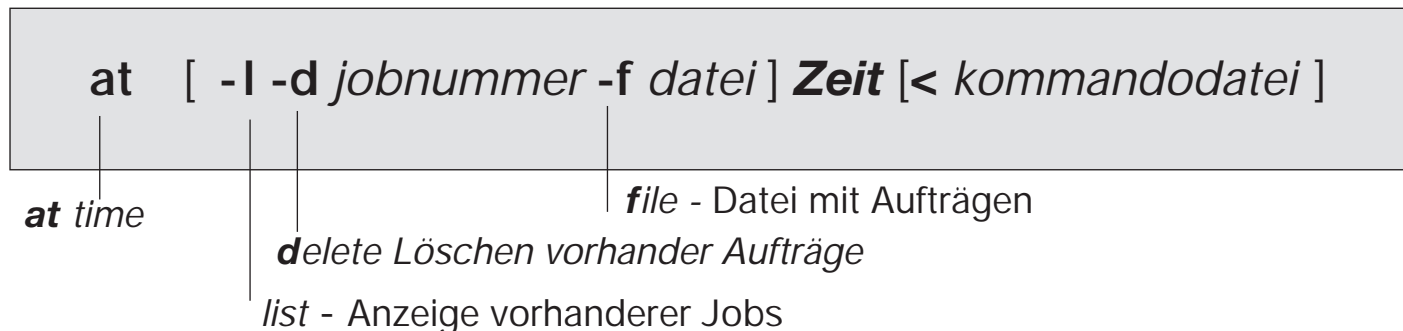
(rechte  
Maustaste)

# Zeitgesteuerte Prozessausführung

- ❑ Über Kommando **at** (**atq**, **batch**, **atrun**)
  - einmalige Jobs
  - Nicht alle Benutzer (/etc/at.allow /etc/at.deny)
- ❑ Über **crontab**
  - sich wiederholende Aufgaben
  - Nur für Systemverwalter - oder spezielle Benutzer (oder Eintrag in /etc/cron.allow oder /etc/cron.deny)
  - Eigene Tabelle über Editor erstellen
  - Genaue Syntax beachten

# At-Kommandos

- ❑ Kommando zu bestimmten Zeiten ausführen: **at**
- ❑ at-Aufträge bearbeiten, verwalten: **atq**
- ❑ at-Aufträge löschen: **atrm *jobnummer***
- ❑ **batch** - Aufträge zu Zeiten niedrigster Systembelastung ausführen
- ❑ **atrun** - alle anstehenden at-Kommandos sofort laufen lassen
- ❑ **atd** - Daemon (**Disk and execution Monitor**) für at-Kommandos
- ❑ Syntax des at-Kommandos



## Hinweise zu at

- ❑ Kommandos am besten mit Pfad angeben (whereis date) bzw. eigene Kommandos
- ❑ at-Auftrag mit <Ctrl+d> anschließen (EOF)
- ❑ Terminalausgabe unter grafischer Oberfläche - ungünstig  
Bei nicht vorhandenem Terminal wird die Ausgabe als mail geschickt:  
„Output from your job (at-Nummer):  
To: undisclosed-recipients: (z.B. Mitteilung von echo)
- ❑ Generell den Daemon stoppen und wieder starten:  
**`/etc/init.d/atd [start] [stop]`**

## Zeitangaben mit at

| Eingabe                   | Bedeutung                          |
|---------------------------|------------------------------------|
| <b>at 13:34</b>           | Heute um 13:24 Uhr ausführen       |
| <b>at 08:00 tomorrow</b>  | Morgen um 8:00 Uhr ausführen       |
| <b>at 13:00 5.6.2002</b>  | Am 5.6.2002 um 13:00 Uhr ausführen |
| <b>at now + 5 minutes</b> | In 5 Minuten ausführen             |
| <b>at now + 5 hours</b>   | In 5 Stunden ausführen             |
| <b>at now + 5 days</b>    | In 5 Tagen ausführen               |
| <b>at now + 5 weeks</b>   | In 5 Wochen ausführen              |
| <b>at midnight</b>        | Mitternacht ausführen              |
| <b>at noon</b>            | Mittags ausführen                  |

# Zeitgesteuerte Arbeiten mit cron

- ❑ Regelmäßig wiederkehrende Arbeiten  
z. B. Sicherungen, Löschen bestimmter Dateien (backup, core-Files etc.)
- ❑ Daemon zum Steuern der crontab-Aufträge: **crond**
- ❑ Systemweite Tabellen /etc/crontab

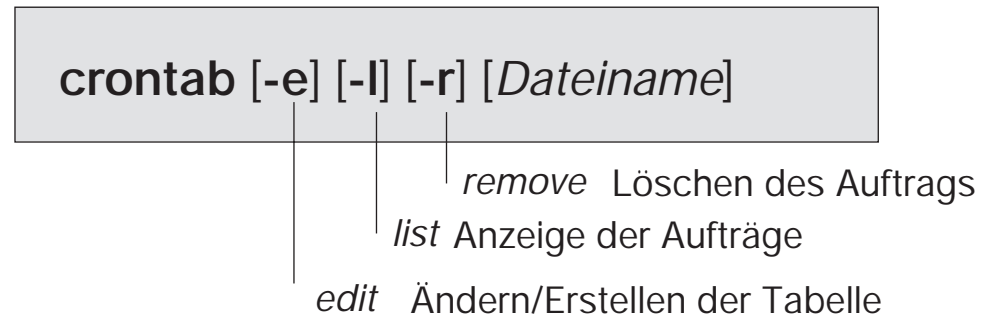
```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
MAILTO=root
# check scripts in cron.hourly, cron.daily, cron.weekly, and cron.monthly
-*/15 * * * * root test -x /usr/lib/cron/run-crons && /usr/lib/cron/run-crons >/dev/null 2>&1
59 * * * * root rm -f /var/spool/cron/lastrun/cron.hourly
14 0 * * * root rm -f /var/spool/cron/lastrun/cron.daily
29 0 * * 6 root rm -f /var/spool/cron/lastrun/cron.weekly
44 0 1 * * root rm -f /var/spool/cron/lastrun/cron.monthly
```

Wochentag (0-7) (0 oder 7 ist Sonntag)

|              |   |            |                              |
|--------------|---|------------|------------------------------|
| Monat 1-12   | } | Jederzeit: | *                            |
| Tag 1-31     |   | Liste:     | 1, 3, 5, 8                   |
| Stunde 0-23  |   | Bereich:   | 9-12                         |
| Minute 01-60 |   | Schritte   | */3 für alle 3 Zeiteinheiten |

# Crontab für den Benutzer

- ❑ Für Benutzer nur erlaubt, wenn eine der beiden Dateien vorhanden ist: /etc/cron.allow bzw. /etc/cron.deny und der Benutzer eingetragen, bzw. nicht ausgeschlossen wurde
- ❑ Übergabe an den crond (Daemon)



- ❑ Gleicher Aufbau der Datei (s. /etc/crontab)
- ❑ Ohne Angabe einer Datei Direkteingabe des Auftrag über Editor

# 17 Hilfe zur Selbsthilfe

- ❑ Ordnung ist das halbe Leben
- ❑ Selbstauskunft
- ❑ Online-Hilfen: man, info und Co.
- ❑ Hilfe im Internet
- ❑ Notfall-CD und Knoppix

# Ordnung ist das halbe Leben $\mathbb{R}$ Rechner-Ordner

- ❑ Ausdruck wichtiger Systemdateien  
/etc/fstab, /etc/passwd, /etc/groups, /etc/hosts  
(Kommando **sitar**)
- ❑ Logbuch, jede Veränderung (Software, Hardware) protokollieren
- ❑ Sicherungsprotokolle, wann wurde was gesichert
- ❑ Plattenbelegung  
(Kommando **hvinfo**,  
**System  $\mathbb{R}$  Überwachung  $\mathbb{R}$  Infozentrum  $\mathbb{R}$  Partitionen**)
- ❑ Netzwerk allgemein (IP-Nummern)
- ❑ Modem-, ISDN-, DSL-Einstellungen  
(Telefonnummern etc.)
- ❑ Hardwareinformationen allgemein  
**System  $\mathbb{R}$  Überwachung  $\mathbb{R}$  Infozentrum**

# Kommandos, die Auskunft geben über das System

| Kommando                | Auskunft  |
|-------------------------|---|
| <b>hostname</b>         | Rechnername   |
| <b>hostid</b>           | Host-ID   |
| <b>domainname</b>       | NIS/YP-Domainname   |
| <b>dnsdomainname</b>    | DNS-Domainname  |
| <b>kernelversion</b>    | Hauptversion des Kernels (2.4, 2.6, ...)                  |
| <b>uname --all</b>      | U. a. Name des Kernels und des Rechners, Kernel-Version   |
| <b>lsmod</b>            | Status der geladenen Kernel-Module                        |
| <b>rpm -q Parameter</b> | Paketinformationen (siehe Seite Seite 355)                |
| <b>route</b>            | IP-Routing-Tabelle  |
| <b>ip Parameter</b>     | Netzwerkinformationen                                     |
| <b>ifconfig</b>         | Status/Einstellungen der Netzwerkschnittstellen           |
| <b>netstat</b>          | Umfangreiche Anzeigemöglichkeiten rund ums Netzwerk       |
| <b>fdisk -l</b>         | Plattengeometrie und Partitionstabelle                    |
| <b>hwinfo</b>           | Umfassende Darstellung der Hardwareinformationen          |
| <b>sitar</b>            | Umfangreiche Darstellung nahezu aller Systeminformationen |

# Online-Hilfen: man, info und Co.

**man** [-k] [*Kapitel-Nr*] [*Kommando*] [*Begriff*]

- k (kriterien) gibt alle möglichen Kommandos aus, zu denen der angegebene Begriff (Schlüsselwort) gefunden (**apropos**)

Kapitel-Nummern:

- 1 Unix/Linux-Kommandos (Utilities) und Shell-Befehle
- 2 Systemaufrufe (Kernel-Funktionen mit ihrer C-Schnittstelle)
- 3 C-Bibliotheksroutinen (Linux-Systembibliotheken)
- 4 Beschreibung der Gerätetreiber und Gerätecharakteristika (zumeist unter /dev)
- 5 Formate spezieller Dateien (Systemdateien, z. B. /etc/fstab und Konfigurationen)
- 6 Spiele – meist nicht sehr ergiebig
- 7 Tabellen, Makropakete und Konventionen (z. B. ASCII-Code)
- 8 Systemverwaltung (zumeist Befehle für root)
- 9 Kernel-Routinen

# SUSE-spezifische Hilfen

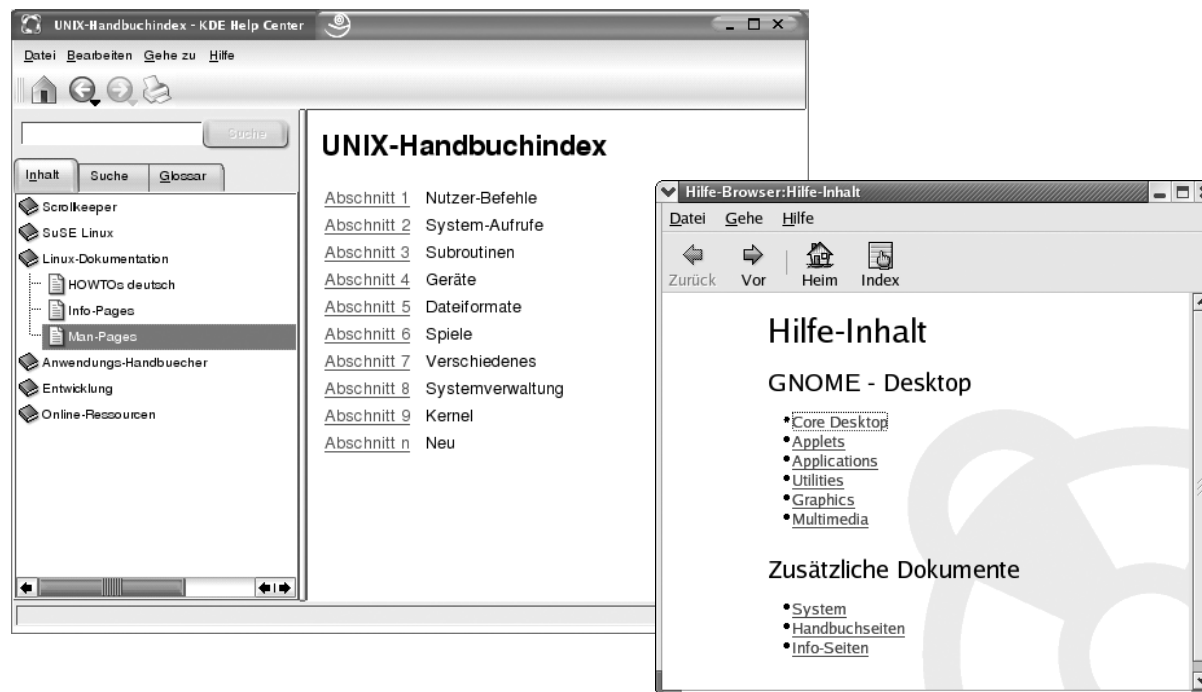
**info** [*Optionen*] *Begriff*

- ❑ info-Seiten  
hierarchisch strukturiert und besitzen Querverweise (oft auch in Deutsch)
- ❑ zusätzlich Html-Aufruf über den Konqueror

**man:cmd** oder **info:cmd**

# Hilfesysteme mit grafischer Oberfläche

SUSE- und KDE-Hilfesystem unter dem KDE-Desktop



## Hilfe im Internet

- ❑ [portal.suse.com/sdb/de/index.html](http://portal.suse.com/sdb/de/index.html)
- ❑ [www.redhat.com/apps/support/knowledgebase/](http://www.redhat.com/apps/support/knowledgebase/)
- ❑ [www.suse.de/de/private/support/online\\_help/howto/index.html](http://www.suse.de/de/private/support/online_help/howto/index.html)

## Online-Foren, Usergroups und Newsgroups

- ❑ [www.linuxforen.de/forums/](http://www.linuxforen.de/forums/)
- ❑ [www.linux.de/groups/](http://www.linux.de/groups/)
- ❑ [groups.google.de/groups?hl=de&lr=&group=comp.os.linux](http://groups.google.de/groups?hl=de&lr=&group=comp.os.linux)

# Notfall-CD und Knoppix

- ❑ ständig gepflegt
- ❑ vollständig im Hauptspeicher und von CD
- ❑ Standardwerkzeuge
- ❑ Hardware-Überprüfung
- ❑ Reparaturen (auch für Windows)

# Wichtige Tastaturkombinationen

| Auswirkung  | Tastaturkombination   |
|---|---|
| Abmelden mit Auswahl für<br>- Sitzung beenden<br>- Rechner ausschalten<br>- Rechner neu starten   | <Strg+Alt+Ent>  |
| Systemüberwachung   | <Strg+Esc>  |
| Befehlseingabe  | <Alt+F2>  |
| Mausemulation   | <Alt+F12>   |
| Aktuelles Fenster schließen   | <Alt+F4>  |
| Fenster schließen (nach Auswahl mit der Maus)   | <Alt+Strg+Esc>  |
| Alle Fenster minimieren   | <Alt+Strg+D>  |
| Fenster verschieben   | <Alt+gedrückte linke<br>Maustaste>  |
| Auf ASCII-Terminal (virtuelle Konsole) wechseln (1 bis 6)<br>Zurück zum grafischen Terminal<br>Für weitere Sitzungen (soweit vorhanden) | <Alt+Strg+F1><br>... <Alt+Strg+F6><br><Alt+Strg+F7><br><Alt+Strg+F8><br>...<Alt+Strg+F12> |
| Bildschirm sperren  | <Alt+Strg+L>  |