

Vorbereitungskurs Informatik - Tag 1

FSI Informatik

Uni Erlangen-Nürnberg

16. Oktober 2007

Wer sind wir?

- Fachschaftsinitiative (kurz FSI) Informatik, IuK und CE
- Was machen wir?
 - Vertretung der studentischen Interessen in Gremien
 - Erstsemestereinführung
 - Bereitstellen von Prüfungsfragen und weiteren Infos
 - Genereller Ansprechpartner für Studenten
 - Sommerfest
 - Was noch so anfällt...
- Wie erreicht man uns?
 - online: <http://fsi.informatik.uni-erlangen.de>
 - per E-Mail fsi@informatik.uni-erlangen.de
 - oder einfach im 2. Stock im Blauen Hochhaus (direkt neben dem CIP) vorbeischaun, Zimmer 02.150
 - online: <http://iuk.uni-erlangen.de/>
 - per E-Mail iuk@stud.informatik.uni-erlangen.de
 - online: www.fsce-erlangen.de
 - per E-Mail fsce@fsce-erlangen.de

Wie schauts im CIP aus?

- CIP-Pools im 1. und 2. Stock des Blauen Hochhauses
- Linux-Arbeitsrechner
- Sunrays
- Drucker
- im RRZE:
 - Scanner
 - Farbdrucker
 - Posterdrucker



Linux - Was ist das?

- eigentlich nur der Kern eines Betriebssystems
- meistens meint man damit eine Zusammenstellung von:
 - Betriebssystem
 - Arbeitsprogrammen
 - Spielen
 - etc.
- diese „Distributionen“ haben eigene Namen und Versionsnummern, z.B.:
 - SuSE
 - Fedora
 - Ubuntu
 - Debian (hier im CIP installiert)

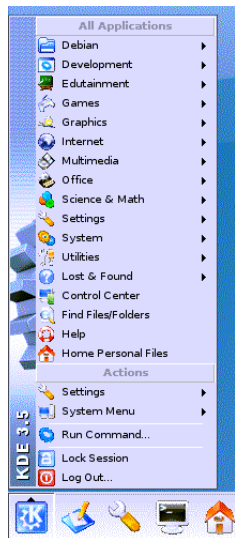
- Window-Manager
- Shell
- weitere Software:
 - Editoren
 - Browser
 - etc.
- diese Komponenten kann man nach eigenen Bedürfnissen selbst zusammenstellen

- bestimmt Aussehen und Verhalten der graphischen Oberfläche
- es existiert ein breites Spektrum:
 - geringer Ressourcenverbrauch
 - minimalistisch
 - nur per Tastatur steuerbar
 - sehr individuell konfigurierbar
 - verspielt
 - Rundum-Paket
 - Grenzen oft fließend

Konfiguration im CIP

Im CIP-Pool kannst du mit dem Befehl `envcfg` deinen Window-Manager auswählen.

- gut geeignet für den Einstieg
- viele Menüpunkte
- Konqueror (Dateimanager und Browser)
- Kontrollzentrum zur Konfiguration
- virtuelle Desktops
- bei vielen Distributionen der Standard



- auf einer Shell kann man Befehle per Tastatur eingeben und ausführen
- unterschiedliche Shells (wie unterschiedliche Window Manager), die gängigsten sind
 - tcsh (Standard im CIP-Pool)
 - **bash** (wird von uns empfohlen)
 - zsh
- Grundfunktionalität gleich
- jede Shell bietet gewisse Komfortfunktionen, z.B.:
 - Tab-Completion
 - Befehlshistory
 - Befehlsaliase
 - Pipes
 - Scripting

- grundlegend für die Arbeit mit dem Dateisystem sind:
 - ls: zeigt Dateien an
 - cd: Verzeichniswechsel
 - mkdir: legt Verzeichnis an
 - cp: kopiert Dateien und Verzeichnisse
 - mv: verschiebt Dateien und Verzeichnisse
 - rm: löscht Dateien und Verzeichnisse
- kleine nützliche Tools
 - grep: sucht im Inhalt von Dateien
 - cat: gibt Textdateien aus
 - less: Anzeige von Text-Dateien mit Scroll-Funktion

ls

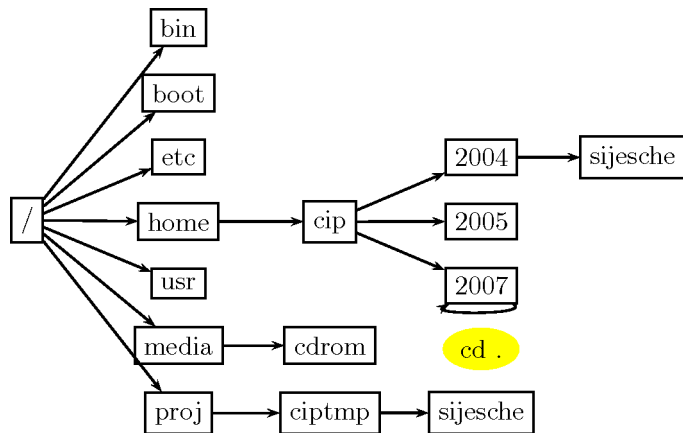
- `ls` listet die Dateien im aktuellen Verzeichnis auf
- `ls verzeichnisname` zeigt den Inhalt des angegebenen Verzeichnisses an
- `ls -l` ausführliches Verzeichnislisting mit Dateigrößen, Rechten, Zeitstempel etc.
- `ls -a` listet auch versteckte Dateien (`.file`) auf

Optionen wie `-l` und `-a` können auch kombiniert werden: `ls -la`

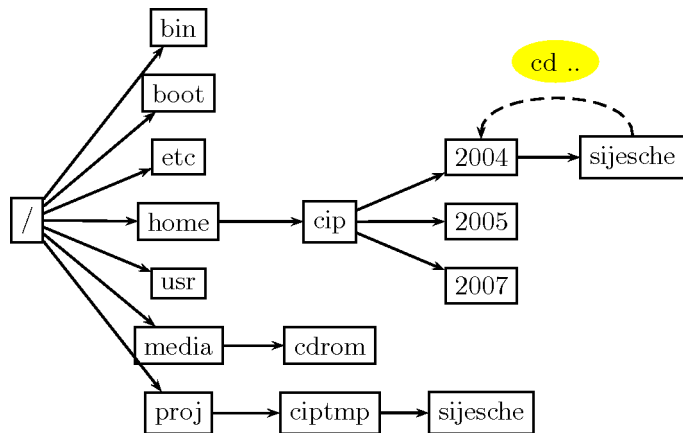
pwd

`pwd` gibt den Pfad des aktuellen Verzeichnisses aus

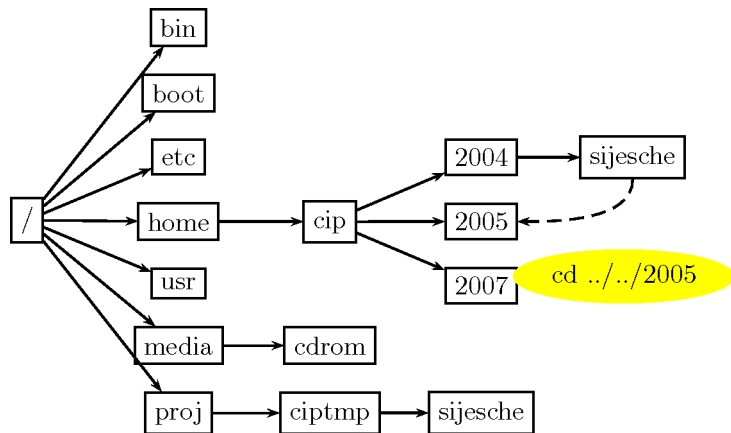
Verzeichnisbaum



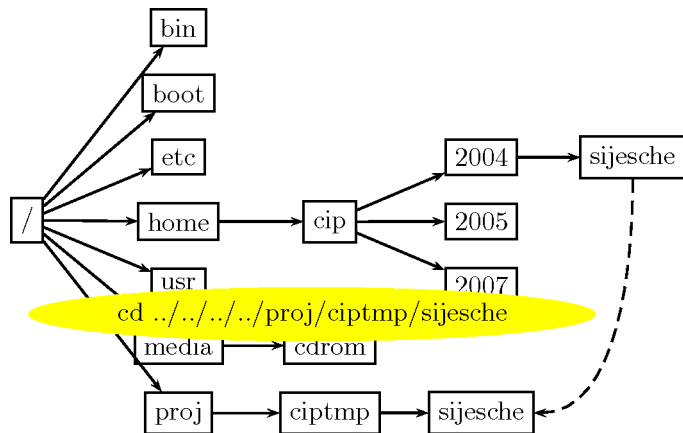
Verzeichnisbaum



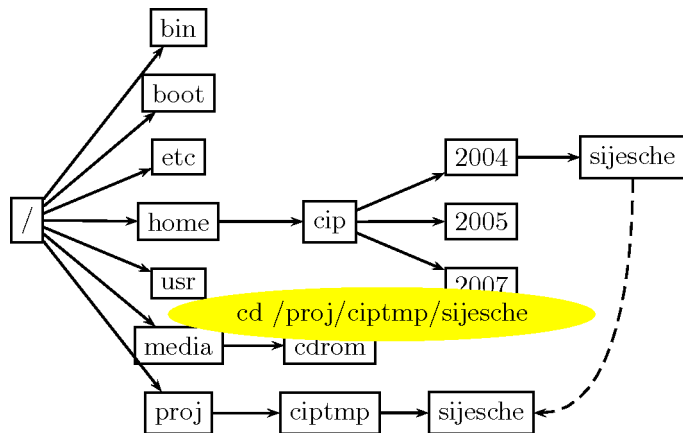
Verzeichnisbaum



Verzeichnisbaum



Verzeichnisbaum



cd

Mit `cd` wechselt man zwischen Verzeichnissen.

| | |
|---------------------------------------|---------------------------------------------------------------------------------------------------|
| <i>Beispiele:</i> <code>cd bin</code> | wechselt in das Unterverzeichnis 'bin' im aktuellen Verzeichnis (<i>relativer Pfadwechsel</i>) |
| <code>cd /bin</code> | geht in das Verzeichnis 'bin' unterhalb vom Root-Verzeichnis (/) (<i>absoluter Pfadwechsel</i>) |
| <code>cd ..</code> | wechselt eine Verzeichnisebene nach oben |
| <code>cd</code> | geht von allen Verzeichnis in das Home-Verzeichnis |
| <code>cd ../testy</code> | wechselt eine Verzeichnisebene nach oben und darin in das Verzeichnis 'testy' |

- jeder User besitzt ein Home-Verzeichnis (/home/cip/2007/userlogin):
 - es steht nur begrenzter Speicherplatz zur Verfügung
 - dort liegen Konfigurationen und Nutzdaten
 - der Inhalt wird täglich gesichert
- mehr Speicherplatz unter /proj/ciptmp/userlogin verfügbar:
 - insgesamt 2 GB
 - wird nicht gesichert!

mkdir

`mkdir foo` `mkdir foo` legt ein Verzeichnis 'foo' im aktuellen Verzeichnis an

rmdir

`rmdir foo` löscht das leere Verzeichnis 'foo' im aktuellen Verzeichnis

cp

cp kopiert Dateien

Aufbau: cp *Quelle Ziel*

Beispiel: cp bsp bspkopie

cp bsp test/

cp -r test/ test2

kopiert die Datei 'bsp' nach
'bspkopie' (im aktuellen Verzeichnis)

kopiert die Datei 'bsp' in das
Verzeichnis 'test'

erstellt eine Kopie des Verzeichnisses
'test' mit dem Namen 'test2'

mv

mv verschiebt Dateien oder benennt sie um

Aufbau: `mv Quelle Ziel`

Beispiel: `mv alt neu`

`mv foo dinge/`

benennt die Datei 'alt' in 'neu' um
(geht auch für Verzeichnisse)
verschiebt die Datei 'foo' aus dem
aktuellen Verzeichnis in das
Verzeichnis 'dinge'

rm

rm löscht Dateien und Verzeichnisse

Beispiel: `rm foo.pdf`

löscht die Datei 'foo.pdf' im
aktuellen Verzeichnis

`rm -r Mails/`

löscht das Verzeichnis 'Mails' und
alle darin enthaltenen Dateien und
Unterverzeichnisse

rm

rm löscht Dateien und Verzeichnisse

Beispiel: `rm foo.pdf`

löscht die Datei 'foo.pdf' im
aktuellen Verzeichnis

`rm -r Mails/`

löscht das Verzeichnis 'Mails' und
alle darin enthaltenen Dateien und
Unterverzeichnisse

Achtung!

rm löscht ohne Nachfrage

- moderne Shells nehmen einem viel Tipparbeit ab, indem sie Namen von Befehlen, Dateien und Verzeichnissen ergänzen
- hierzu tippt man den Anfang des Namens und dann <TAB>
- wenn nicht eindeutig, Liste von Alternativen mit <TAB> <TAB>

Beispiel

```
faii00a [~]> ls
```

```
Desktop  core  folien_vorkurs_2007_tag1.pdf
```

```
faii00a [~]> xpdf fo<TAB>
```

```
faii00a [~]> xpdf folien_vorkurs_2007_tag1.pdf
```

- die Shell merkt sich die zuletzt eingegebenen Befehle
- aufrufen der letzten Befehle mit den Cursortasten hoch/runter
- Ausgabe der letzten Befehle mit `history`

lpr

lpr druckt ein Dokument aus

Beispiel: `lpr foo.pdf`

`lpr -P<DRUCKER> foo.pdf`

druckt die Datei foo.pdf auf dem Standarddrucker aus
druckt die Datei foo.pdf auf dem Drucker mit den Namen <DRUCKER> aus

In fast jedem CIP-Pool stehen Drucker. Die Namen der Drucker sind am Gerät abzulesen.

pr_acct

`pr_acct` Zeigt das aktuelle Druckguthaben an

lpq - Druckerwarteschlange

lpq

lpq zeigt die aktuelle Druckerwarteschlange an

```
faiu00a [~]> lpq
```

```
Printer: ps2acip@faiu00a (dest ps2acip@faiu02.informatik.uni-e
```

```
Queue: no printable jobs in queue
```

```
Status: job 'sijojord@faiu00a+632' saved at 16:21:37.220
```

```
Printer: ps2acip@faiu02
```

```
Queue: no printable jobs in queue
```

```
Server: no server active
```

```
Status: job 'sihsuern@faiu08+331' saved at 23:19:42.548
```

```
Filter_status: (of) done at 23:19:42.543
```

| Rank | Owner/ID | Pr/Class | Job Files |
|------|---------------------|----------|------------------------|
| done | sihsuern@faiu08+331 | A | 331 /tmp/kde-sihsuern, |

lprm

lprm löscht Druckaufträge aus der Druckerwarteschlange

Beispiel: lprm

löscht den zu letzten
Druckauftrag

lprm -a all

löscht alle eigenen
Druckaufträge aus allen
Druckerwarteschlangen

Es gibt eine Vielzahl von Programmen für Linux um die unterschiedlichsten Dateiformate zu betrachten, z.B.:

- gedit, scite, kwrite, gvim, ... für Textdateien
- gv, ggv für Postscript-Dateien
- acroread, xpdf, kpdf, evince für PDF-Dateien
- gqview, display für alle möglichen Bilddateien
- mplayer, xine, vlc für Videos

- viele Shell-Befehle arbeiten mit Standard-Ein- und Ausgabe
- oft ist die Ausgabe eines Programmes gleichzeitig wieder Eingabe für ein weiteres
- gebräuchlich sind:
 - `<`, um die Eingabe aus einer Datei lesen zu lassen
 - `|` (sprich: Pipe), um die Ausgabe des einen Programms als Eingabe des anderen zu verwenden
 - `>`, um die Ausgabe in eine Datei zu schreiben
- Beispiel: Suche alle Zeilen, in denen „backup“ vorkommt in der Datei `/proj/ciptmp/README`
 - `cat /proj/ciptmp/README | grep backup`
 - `grep backup < /proj/ciptmp/README`