

# IT-Infrastruktur

SS 2015

**Hans-Georg Eßer**  
Dipl.-Math., Dipl.-Inform.

## Foliensatz F:

- Zusammenfassung der Vorlesung

v1.2, 2015/06/15

15.06.2015

IT-Infrastruktur, SS 2015, Hans-Georg Eßer

Folie F-1

## Dieser Foliensatz

Vorlesungsübersicht

Seminar

Wiss. Arbeiten

Datenformate und Wandlung

PC als Arbeitsplatz

Ergonomie und Arbeitsschutz

Rechnerstrukturen

Zentrale / verteilte IT-Infrastrukturen

Zusammenfassung

Folien F

15.06.2015

IT-Infrastruktur, SS 2015, Hans-Georg Eßer

Folie F-2

## A: Datenformate und Wandlung (1)

- Information vs. Daten
- Bits, Bytes, dual/oktal/hex
- ASCII, Unicode (UTF-8, UTF-16)
- Digitalisierung:
  - rastern (diskretisieren, sampeln) und
  - quantisieren (Wertebereich einschränken)
- Kompression (verlustfrei/verlustbehaftet)
- Prüfsummen, Fehlerkorrektur

15.06.2015

IT-Infrastruktur, SS 2015, Hans-Georg Eßer

Folie F-3

## A: Datenformate und Wandlung (2)

- Mark-up: HTML/CSS, LaTeX, Wiki,
- XML, XSLT, DTD, XHTML, DocBook, Open Document Format (XML)
- Rastergrafiken (JPG vs. PNG), Vektorgrafiken
- PostScript, PDF
- MPEG, Frames
- Kommunikation
  - Client / Server; HTTP, FTP, SMTP
  - RPC, asynchronous RPC

15.06.2015

IT-Infrastruktur, SS 2015, Hans-Georg Eßer

Folie F-4

- ~~Datenträgeraustauschformat (DTaus)~~
- Applikationsformate (CAD, GIS, DTP)
- Archive und Software-Pakete
  - zip, tar, gz, tar.gz (plattform-übergreifend?)
- ~~Linux: RPM, Debian;~~
  - Pakete mit Metadaten
  - Abhängigkeiten, Konflikte
  - Repositories: Abh. auflösen, Upgrade
- ~~Windows: MSI, OS X: DMG mit \*.app-Ordern~~

- ~~Exkurs: LaTeX, BibTeX~~
- Statistik: Programmiersprache R
- Numerik: GNU Octave
- Computer-Algebra-Systeme, Wolfram Alpha
- GIS: Mark-up-Sprachen, Nielsen-Gebiete
- Versionsverwaltung, Mercurial (hg), Klonen, Merge-Operation

- „Gebrauchstauglichkeit“
- Nutzungskontext(e), Zielgruppe(n)
- effektiv, effizient, zufrieden stellend
- Kriterien
  - Aufgabenangemessenheit
  - Selbstbeschreibungsfähigkeit
  - Steuerbarkeit
  - Erwartungskonformität
  - Fehlertoleranz
  - Individualisierbarkeit
  - Lernförderlichkeit

- Beispiele:
  - ~~Microsoft Ribbons~~
  - ~~Seriennummern-Eingabe~~
    - Secure Shell (ssh, scp; Optionen -p, -P)
  - Kriterien auch auf Webdesign anwendbar
  - ~~Arbeitsschutz~~

- Universalrechner, von-Neumann vs. Harvard
- ISA (Instruction Set Architecture)
  - Maschinenbefehle, Register
  - Adressierungsarten, Interruptbehandlung
  - 1-/2-/3-Adress-Maschinen bzw. -Befehle
  - Spezialregister (IP, SP, Status)
- Load, Store, Push, Pop, arithm. Operationen  
→ RISC, CISC

- Pipelining
  - 5-stufige RISC-Pipeline (Fetch, Decode, Execute, Memory Access, Write-back)
  - 6-stufige CISC-Pipeline (Fetch, Decode, Calculate Operands, Fetch Operands, Execute, Write-back)
  - Pipeline-Hemmnisse
    - strukturell (Speicherzugriff bei **M**, **F**)
    - Datenabhängigkeiten (RAW-Konflikt)
    - ablauf-bedingt (bedingter Sprung; Sprungvorhersage)

- Superskalare Architekturen
  - mehrere Execute-Einheiten, nach Aufgaben getrennt (z. B. FPU, Int, MUL)
  - abhängige vs. unabhängige Pipelines
  - Datenabhängigkeiten (RAW, WAR, WAW; RAR)
  - Abhängigkeitsgraph (ohne RARs, transitive Abh.)
  - Reorder-Buffer
    - Platz im Buffer, Anzahl der Ausführ-Einheiten (pro Kategorie), Anzahl der pro Takt zuteilbaren (issue) und bestätigten (commit) Befehle
- ~~Register Renaming~~

- Parallele Systeme
- Programm parallelisieren, Speed-up  
→ Amdahls Gesetz:  $S(n) \rightarrow 1 / \alpha$
- Arten der Parallelität (Bit-Level, Instruction-Level, Daten/Schleifen-Parall., Task-Parall.)
- Verteilte Systeme
- Client / Server, RPC, Reverse Proxy, Asynchronous RPC
- Cluster, Super Computer (HPC, MPI), Grid

- Cloud Computing
  - skalierbar, elastisch, abrechenbar, zentralisiert
  - Virtualisierung
  - public / private / community / hybrid cloud
  - Cloud-Vorteile: pay as you go, keine Investitionen
  - Varianten:
    - Software as a Service (SaaS),
    - Platform as a Service (PaaS),
    - Infrastructure as a Service (IaaS)

- Letzte Fragen?

- Technische Umsetzung
  - Verteilte Konfiguration mit ZooKeeper
  - ~~• Map und Fold (funktionale Programmierung)~~
  - ~~• Software Entwicklung mit MapReduce~~
  - Verteilte Dateisysteme (GPFS, GoogleFS statt NFS)