



Script generated by TTT

Title: Einf_HF (05.05.2014)

Date: Mon May 05 14:14:42 CEST 2014

Duration: 91:17 min

Pages: 34

Das World Wide Web (WWW bzw. Web) ist ein über das Internet abrufbares Informationssystem, basierend auf dem Hypertext-Ansatz.

[Hypertext](#)

[Einführung in HTML](#)

[Cascading Style-Sheets \(CSS\)](#)

Generated by Targeteam



Beispiel



Einführung in HTML



```
<html>
<head>
  <title>Ziel der Vorlesung</title>
</head>
<body>
<p>Diese Vorlesung gibt eine Einführung in die Informatik. Sie behandelt
Themen wie</p>
<ul>
  <li>Algorithmen und Datenstrukturen</li>
  <li>Programmierung von Softwaresystemen</li>
  <li>Datenbanken</li>
  <li>Rechnernetze</li>
</ul>
</body>
</html>
```

Generated by Targeteam

HTML dient in erster Linie dazu, textuelle Information zu repräsentieren. Der Text lässt sich strukturieren, i.w. in Paragraphen, Listen und Tabellen.

ein ausführliches Tutorial stellt [selfhtml](#) bereit.

HTML Befehle bestehen fast immer aus einem Starttag und einem Endtag.

[Beispiel](#)

[Struktur-Tags](#)

[Listen](#)

[Tabellen](#)

[Grafiken](#)

[Hyperlinks](#)

Generated by Targeteam



HTML dient in erster Linie dazu, textuelle Information zu repräsentieren. Der Text lässt sich strukturieren, i.w. in Paragraphen, Listen und Tabellen.

ein ausführliches Tutorial stellt **selfhtml** bereit.

HTML Befehle bestehen fast immer aus einem Starttag und einem Endtag.

Beispiel

Struktur-Tags

Listen

Tabellen

Grafiken

Hyperlinks

Generated by Targteam



```

<table>
<tr>
  <th>Kopfzeile / 1. Spalte</th>
  <th>Kopfzeile / 2. Spalte</th>
  <th>Kopfzeile / 3. Spalte</th>
</tr>
<tr>
  <td>1.Zeile / 1. Spalte</td>
  <td>1.Zeile / 2. Spalte</td>
  <td>1.Zeile / 3. Spalte</td>
</tr>
<tr>
  <td>2.Zeile / 1. Spalte</td>
  <td>2.Zeile / 2. Spalte</td>
  <td>2.Zeile / 3. Spalte</td>
</tr>
</table>

```

<th>..	<th>..	<th>..	</tr>
</th>	</th>	</th>	
<td>..	<td>..	<td>..	</tr>
...	



Tabellen erlauben die Anordnung von Paragraphengruppen in einem zweidimensionalen Gitter.

<table> </table>: Klammerung der gesamten Tabelle.

<tr> </tr>: Klammerung einer Tabellenzeile.

<td> </td>: Klammerung einer Tabellenzelle.

<th> </th>: Klammerung einer Kopfzelle der Tabelle.

Beispiel

Attribute in <table> bzw. <td> zur Gestaltung von Tabellen, z.B.

<table border="..">: Dicke des sichtbaren Rahmens.

<table cellpadding="..">: Abstand des Zelleninhalts vom Rand der Tabellenzelle.

<table width="..">: bestimmt die Breite der gesamten Tabelle.

<td align=".." valign="..">: horizontale und vertikale Ausrichtung des Inhalts innerhalb einer Zelle.

Generated by Targteam

<th>..	<th>..	<th>..	</tr>
</th>	</th>	</th>	



```

<table>
<tr>
  <th>Kopfzeile / 1. Spalte</th>
  <th>Kopfzeile / 2. Spalte</th>
  <th>Kopfzeile / 3. Spalte</th>
</tr>
<tr>
  <td>1.Zeile / 1. Spalte</td>
  <td>1.Zeile / 2. Spalte</td>
  <td>1.Zeile / 3. Spalte</td>
</tr>
<tr>
  <td>2.Zeile / 1. Spalte</td>
  <td>2.Zeile / 2. Spalte</td>
  <td>2.Zeile / 3. Spalte</td>
</tr>
</table>

```

<th>..	<th>..	<th>..	</tr>
</th>	</th>	</th>	
<td>..	<td>..	<td>..	</tr>
...	



Tabellen erlauben die Anordnung von Paragraphengruppen in einem zweidimensionalen Gitter.
 <table> </table>: Klammerung der gesamten Tabelle.
 <tr> </tr>: Klammerung einer Tabellenzeile.
 <td> </td>: Klammerung einer Tabellenzelle.
 <th> </th>: Klammerung einer Kopfzelle der Tabelle.

Beispiel

Attribute in <table> bzw. <td> zur Gestaltung von Tabellen, z.B.

- <table border="..">: Dicke des sichtbaren Rahmens.
- <table cellpadding="..">: Abstand des Zellinhalts vom Rand der Tabellenzelle.
- <table width="..">: bestimmt die Breite der gesamten Tabelle.
- <td align=".." valign="..">: horizontale und vertikale Ausrichtung des Inhalts innerhalb einer Zelle.

Generated by Targeteam



Sie sind das wesentliche Element, um Querverweise zwischen HTML-Dokumenten zu definieren. Hyperlinks werden im Text eingebettet.

Markieren des Verweiszziels

```
<a name="nameziel">Text</a> und hier weiterer Text
```

Verweise innerhalb des gleichen Dokuments

```
<a href="#nameziel">Text des Hyperlinks</a> und hier weiterer Text
```

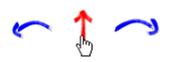
Verweise auf andere lokale Dokumente

```
<a href="Dateiname#nameziel">Text des Hyperlinks</a> und hier weiterer Text
```

Verweise auf andere nicht lokale Dokumente

```
<a href="http://servername/pfad#nameziel">Text des Hyperlinks</a> und hier weiterer Text
```

Generated by Targeteam



HTML dient in erster Linie dazu, textuelle Information zu repräsentieren. Der Text lässt sich strukturieren, i.w. in Paragraphen, Listen und Tabellen.

ein ausführliches Tutorial stellt **selfhtml** bereit.

HTML Befehle bestehen fast immer aus einem Starttag und einem Endtag.

[Beispiel](#)

[Struktur-Tags](#)

[Listen](#)

[Tabellen](#)

[Grafiken](#)

[Hyperlinks](#)

Generated by Targeteam



HTML dient in erster Linie dazu, textuelle Information zu repräsentieren. Der Text lässt sich strukturieren, i.w. in Paragraphen, Listen und Tabellen.

ein ausführliches Tutorial stellt **selfhtml** bereit.

HTML Befehle bestehen fast immer aus einem Starttag und einem Endtag.

[Beispiel](#)

[Struktur-Tags](#)

[Listen](#)

[Tabellen](#)

[Grafiken](#)

[Hyperlinks](#)

Generated by Targeteam





Festlegung der Formateigenschaften von HTML-Tags

interpretieren durch den Browser zum Formatieren und Positionieren von HTML-Elementen auf dem Nutzerrechner.

Einbettung von CSS-Formaten in die Web-Seite oder als separate Datei, die von Web-Seite referenziert wird.

Eigenschaften werden mit Hilfe von Regeln spezifiziert, eine Regel besteht aus Selektor, der Eigenschaft sowie dem ihr zugewiesenen Wert.

Beispiel

```

BODY { font-size: 20px;
       font-family: Helvetica;
       margin-left: 0.5em}

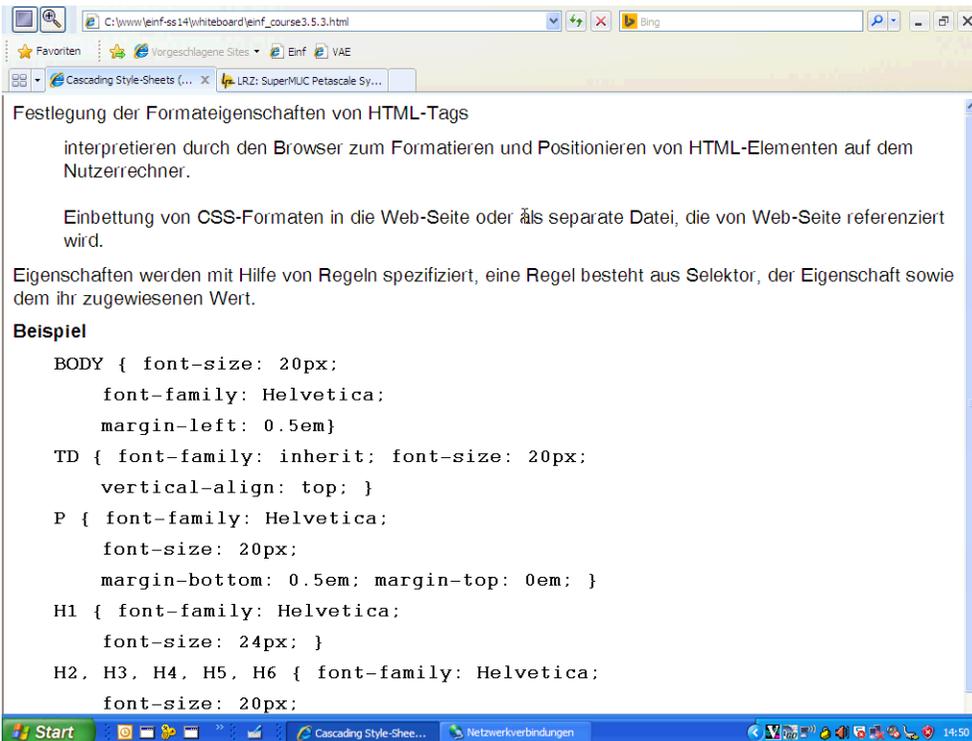
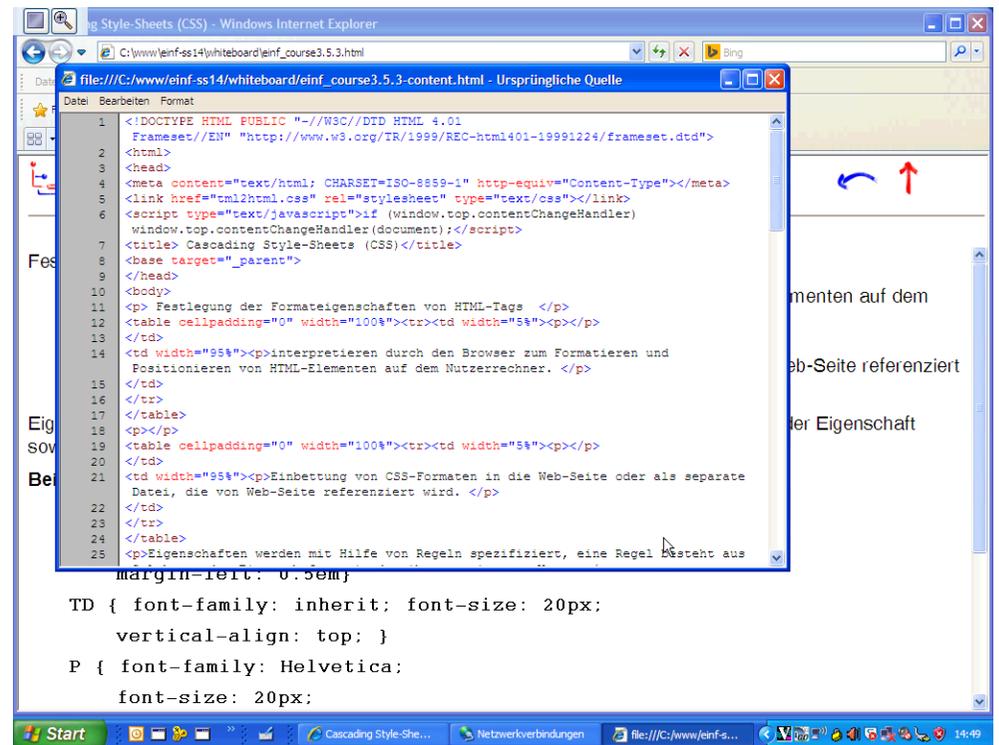
TD { font-family: inherit; font-size: 20px;
     vertical-align: top; }

P { font-family: Helvetica;
    font-size: 20px;
    margin-bottom: 0.5em; margin-top: 0em; }

H1 { font-family: Helvetica;
     font-size: 24px; }

H2, H3, H4, H5, H6 { font-family: Helvetica;
                     font-size: 20px;
                     margin-bottom: 0.5em; margin-top: 0.5em; }

```



Das World Wide Web (WWW bzw. Web) ist ein über das Internet abrufbares Informationssystem, basierend auf dem Hypertext-Ansatz.

Hypertext

Einführung in HTML

Cascading Style-Sheets (CSS)



- Fragestellungen des Abschnitts:
 - Aus welchen (Hardware-)Elementen setzt sich ein Rechner zusammen?
 - Wie kommunizieren die einzelnen Komponenten eines Rechners?
 - Wie sieht die Schnittstelle zwischen Hardware und Software aus (d.h. Maschinenbefehle)?
 - Wie werden Zahlen, Text, Bilder, und Töne intern dargestellt?

Aufbau eines Rechners

Maschinenbefehle

Befehlszyklus

Interdarstellung von Information



Hardware eines Computersystems: Rechner und Peripheriegeräte, z.B. Tastatur, Maus, Bildschirm, Drucker, Scanner. Sehr unterschiedliche Leistungsfähigkeit.

Beispiel eines Supercomputers



Generated by Targetteam

Struktureller Aufbau eines Rechners

Generated by Targetteam

http://www.lrz.de/services/compute/supermuc/

SuperMUC Petascale System

System Status:

- Fat node island:
 - Status: Up and Running
- Thin node islands:
 - Status: Up and Running
- Recent Messages
- Extended Status (node usage, nodes becoming free, job sizes)

System

- System Description of SuperMUC

Contact and Support

- Contact and Support and Servicedesk

Please note: Due to construction work on the northern section of the U6 as well as due to road construction work, reaching our location in Garching will be affected in the period from May 12 to September 15, 2014. For more information, please refer to our notes.

Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities

NEWS FAQ SERVICES SERVICEDESK ID-PORTAL

SuperMUC Petascale System

System Status:

- Fat node island:
 - Status: Up and Running
- Thin node islands:
 - Status: Up and Running
- Recent Messages
- Extended Status (node usage, nodes becoming free, job sizes)

System

- System Description of SuperMUC

Contact and Support

- Contact and Support and Servicedesk

Please note: Due to construction work on the northern section of the U6 as well as due to road construction work, reaching our location in Garching will be affected in the period from May 12 to September 15, 2014. For more information, please refer to our notes.

Search | Kontakt | Impressum | Datenschutz

lrz Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities

NEWS FAQ SERVICES SERVICEDESK ID-PORTAL

Home Page
Wir über uns (de)
Services (de)
Fragen, Beratung und Unterstützung (de)
Forschung (de)
Andere im WWW (de)
SuperMUC
www.lrz.de / services / compute / supermuc / user_affairs
Access and Login
Contact and Support
Education and Training
File Systems
HPC Cans
Interactive and Batch Jobs
Manuals
Parallelization
Program Development
Project Proposal
Recent Messages
Software
Tuning and Optimization
Usage
User Affairs
Acknowledgement

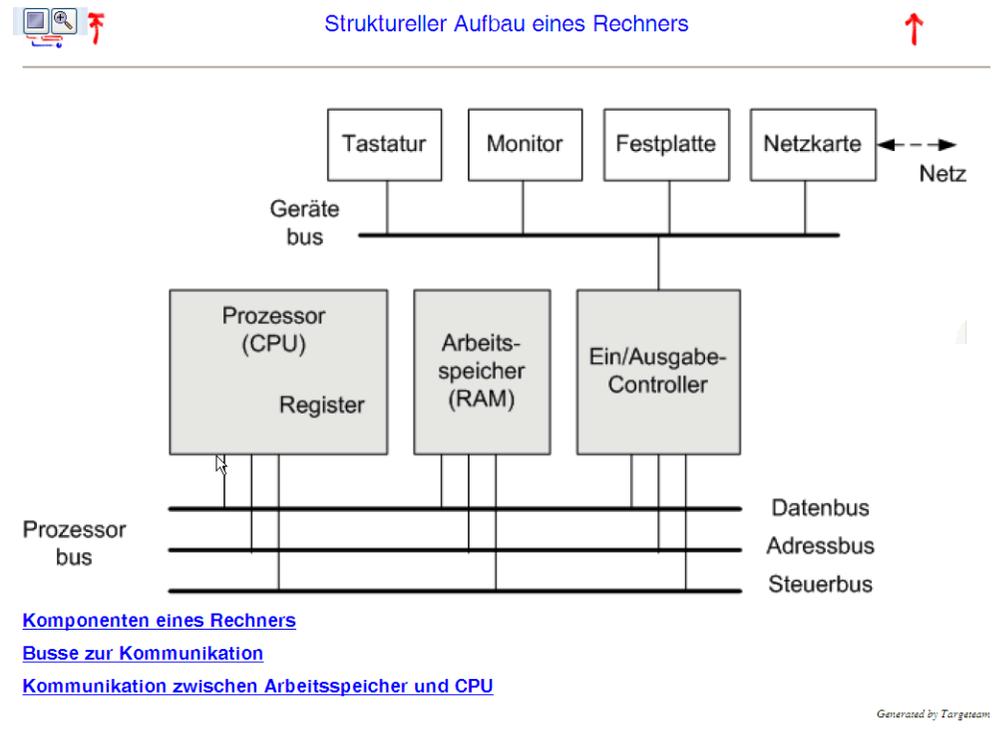
User Affairs, Password, Upload

- Change Password
- User data, budgets, etc.
- Research Allocations Request Form (Online Project Proposal)
- Modification of Data of an Existing Project
- Upload Status, Reports, and Papers
- Status Report Template
- Acknowledging the resources and support provided by LRZ

Informations for ...

- University Staff
- Students
- Supercomputer Users
- Grid Users
- Institutions
- Network Administrators
- System Administrators
- Users with Handicap
- Press

Please note: Due to construction work on the northern section of the U6 as well as due to road construction work, reaching our location in Garching will be affected in the period from May 12 to September 15, 2014. For more information, please refer to our notes.



Komponenten eines Rechners

Im Prozessor: alle Rechen- und Umformvorgänge und die Ablaufsteuerung der Programme.

moderne Prozessoren bestehen aus mehreren Kernen

Register sind spezielle Speicherzellen im Prozessor.

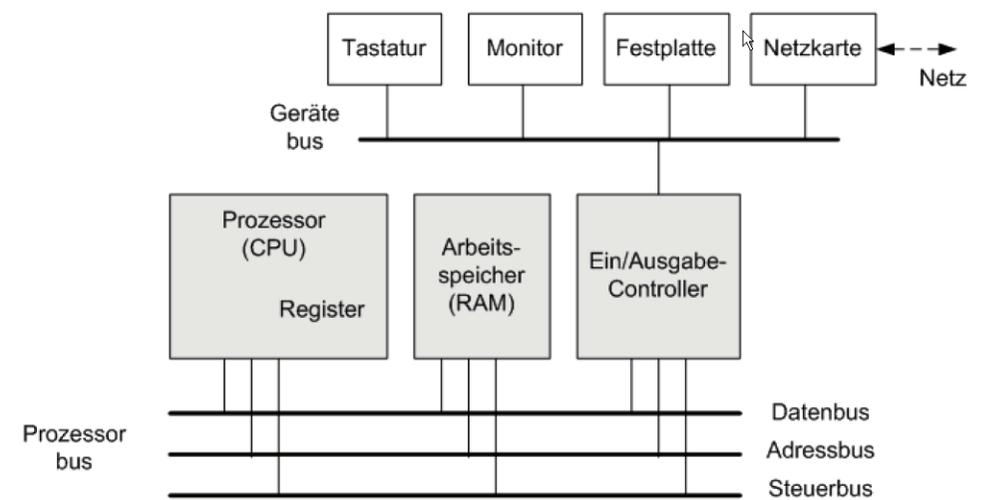
Arbeitsspeicher (ASP; "main memory"): direkt adressierbare Speicherzellen der Größe 8 Bit zur Ablage von Programmen und binären Daten.

Ein/Ausgabe-Controller: setzen Nachrichten auf dem Gerätebus für Ein/Ausgabe-Geräte um, stellen Eingaben der Geräte am Bus bereit. Anschluss von Graphikkarten, Netzwerkkarten etc über den Gerätebus.

Den Ablauf der Informationsverarbeitungsvorgänge im Rechner steuern Programme.

Generated by Targeteam

Struktureller Aufbau eines Rechners





Rechnerkomponenten kommunizieren über **Prozessorbus** und **Gerätebus**

Prozessorbus: schnelle Übertragung

Gerätebus (z.B. PCI): etwas langsamer, da Geräte langsamer als CPU oder Arbeitsspeicher.

Der Prozessorbus unterteilt sich in

Adressbus: zur Übergabe einer Adresse (32 oder 64 Bit)

Datenbus: zur Übermittlung von Daten (32 oder 64 Bit)

Steuerbus: mehrere Steuerleitungen

Generated by Targteam



CPU und Arbeitsspeicher arbeiten nicht synchron \Rightarrow Koordination notwendig.

Beispiel: Übertragung von Daten aus dem Arbeitsspeicher zum Prozessor (CPU).

1. CPU legt die ASP-Adresse des gewünschten Datenelements auf den Adressbus.
2. CPU legt das Signal "Lesen" auf eine bestimmte Leitung im Steuerbus.
3. CPU legt das Signal "Adresse gültig" auf eine bestimmte Leitung im Steuerbus.
4. ASP ist passiv, "hört" aber auf Adress- und Steuerbus mit.
5. ASP erkennt, dass eine Adresse gültig ist und prüft, ob er mit dieser Adresse gemeint ist. Falls ja, führe Schritte 6-12 aus.
6. ASP prüft das Transportrichtungssignal im Steuerbus (es erkennt das Signal "Lesen").
7. ASP sucht die adressierte Speicherzelle und holt die Daten.
8. ASP legt die (adressierten) Daten auf den Datenbus.
9. ASP legt das Signal "Daten bereit" auf eine bestimmte Leitung im Steuerbus.
10. CPU hat gewartet, bis das Signal "Daten bereit" erscheint und übernimmt Daten vom Bus in das Zielregister.
11. CPU legt das Signal "Daten übernommen" auf eine bestimmte Leitung im Steuerbus.
12. ASP deaktiviert daraufhin den Datenbus und das Signal "Daten bereit".
13. Sobald das Signal "Daten bereit" vom ASP weggenommen wird, deaktiviert die CPU die von ihm verwendeten Adress- und Steuerleitungen.

Solcher Ablauf aus Aktions- und Kommunikationsschritten zwischen Kommunikationspartnern heißt "Protokoll".

Der oben geschilderte Ablauf heißt **Busprotokoll**.



Beispiel: Übertragung von Daten aus dem Arbeitsspeicher zum Prozessor (CPU). *Wann ist der gesamte Zeit*

1. CPU legt die ASP-Adresse des gewünschten Datenelements auf den Adressbus.
2. CPU legt das Signal "Lesen" auf eine bestimmte Leitung im Steuerbus.
3. CPU legt das Signal "Adresse gültig" auf eine bestimmte Leitung im Steuerbus.
4. ASP ist passiv, "hört" aber auf Adress- und Steuerbus mit.
5. ASP erkennt, dass eine Adresse gültig ist und prüft, ob er mit dieser Adresse gemeint ist. Falls ja, führe Schritte 6-12 aus.
6. ASP prüft das Transportrichtungssignal im Steuerbus (es erkennt das Signal "Lesen").
7. ASP sucht die adressierte Speicherzelle und holt die Daten.
8. ASP legt die (adressierten) Daten auf den Datenbus.
9. ASP legt das Signal "Daten bereit" auf eine bestimmte Leitung im Steuerbus.
10. CPU hat gewartet, bis das Signal "Daten bereit" erscheint und übernimmt Daten vom Bus in das Zielregister.
11. CPU legt das Signal "Daten übernommen" auf eine bestimmte Leitung im Steuerbus.
12. ASP deaktiviert daraufhin den Datenbus und das Signal "Daten bereit".
13. Sobald das Signal "Daten bereit" vom ASP weggenommen wird, deaktiviert die CPU die von ihm verwendeten Adress- und Steuerleitungen.

Solcher Ablauf aus Aktions- und Kommunikationsschritten zwischen Kommunikationspartnern heißt "Protokoll".

Der oben geschilderte Ablauf heißt **Busprotokoll**.

Generated by Targteam



• Fragestellungen des Abschnitts:

- Aus welchen (Hardware-)Elementen setzt sich ein Rechner zusammen?
- Wie kommunizieren die einzelnen Komponenten eines Rechners?
- Wie sieht die Schnittstelle zwischen Hardware und Software aus (d.h. Maschinenbefehle)?
- Wie werden Zahlen, Text, Bilder, und Töne intern dargestellt?

Aufbau eines Rechners

Maschinenbefehle

Befehlszyklus

Interdarstellung von Information

Generated by Targteam



Darstellung in Maschinensprache

```

0100 LOAD 0118 -- lade Inhalt der Speicherzelle 118 in CPU
0102 STORE 0116
0104 LOAD 0114
0106 JUMPZERO 011a
0108 SUB 0118
010a STORE 0114
010c LOAD 0116
010e ADD 0116
0110 STORE 0116
0112 JUMP 0104
0114 #2 -- Wert von x, d.h. die Zahl 2
0116 #0
0118 #1 -- Wert von y, d.h. die Zahl 1
011a STOP

```

Darstellung in Hochsprache

```

y = 1;
while (x!=0) {
  x = x - 1;
  y = y + y;
}

```



Darstellung in Maschinensprache

```

0100 LOAD 0118 -- lade Inhalt der Speicherzelle 118 in CPU
0102 STORE 0116
0104 LOAD 0114
0106 JUMPZERO 011a
0108 SUB 0118
010a STORE 0114
010c LOAD 0116
010e ADD 0116
0110 STORE 0116
0112 JUMP 0104
0114 #2 -- Wert von x, d.h. die Zahl 2
0116 #0
0118 #1 -- Wert von y, d.h. die Zahl 1
011a STOP

```

Darstellung in Hochsprache

```

y = 1;
while (x!=0) {
  x = x - 1;
  y = y + y;
}

```

x ist ein Wert aus IN
Vergleich != ≙ ungleich
x []
y = y + y

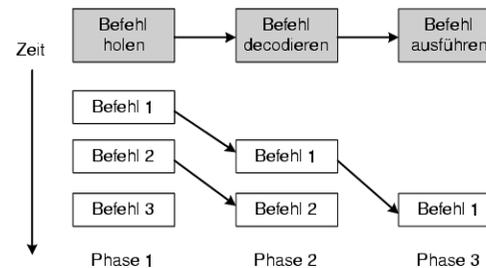


Einfache Kommandos, die die CPU ausführen kann; Setzen sich zusammen aus Operationsteil und Operandenteil (Adressteil).

Befehlsvorrat

Beispielprogramm in Maschinensprache (Assembler)

Bearbeitung jeden Befehls in mehreren Phasen. Überlappende Verarbeitung. Quasi-parallele Ausführung mehrerer Maschinenbefehle.

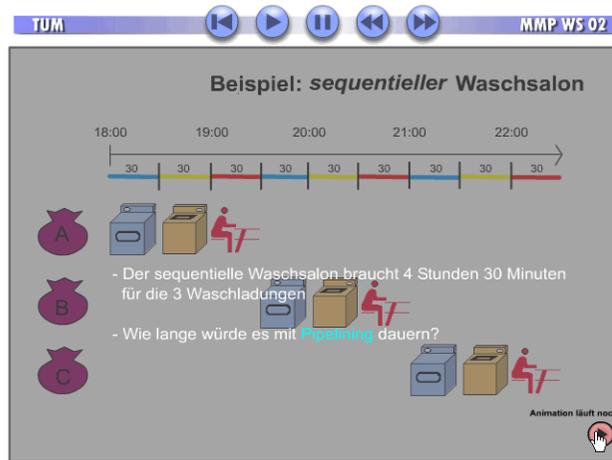


Pipelining Animation



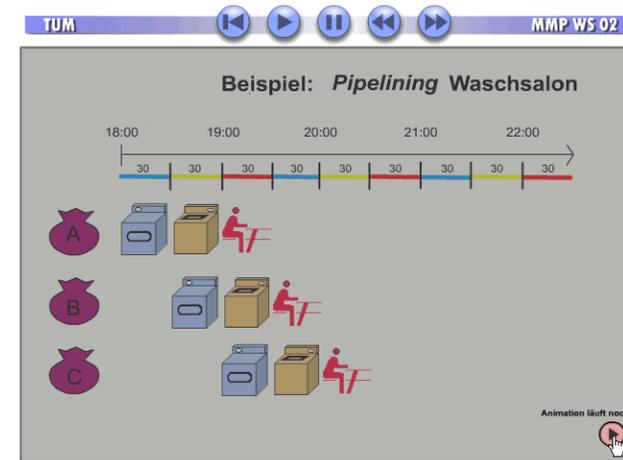
w' 1.2

w' 1.2



wir 1.6

Generated by Targeteam



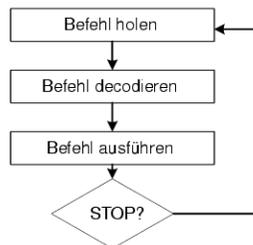
wir 1.6

Generated by Targeteam



Ausführung eines Maschinenbefehls: festes Schema (Bereitstellung, Bereitstellen der Operanden, Befehl entschlüsseln, Ausführung). Dieses Schema heißt Befehlszyklus.

Sequentielle Bearbeitung



kein ausführbares Programm ⇒ Ausführung von NOP ("No Operation").

Fließband Bearbeitung (Pipelining)

wir 1.6

Generated by Targeteam